AD-A241 377

(2)

# NAVAL POSTGRADUATE SCHOOL
## Monterey, California

DTIC
ELECTE
OCT 03 1991
S D
D

# THESIS

FUNCTIONAL SPECIFICATION
FOR A
GENERIC C3I WORKSTATION

by

Steven E. Anderson

September, 1990

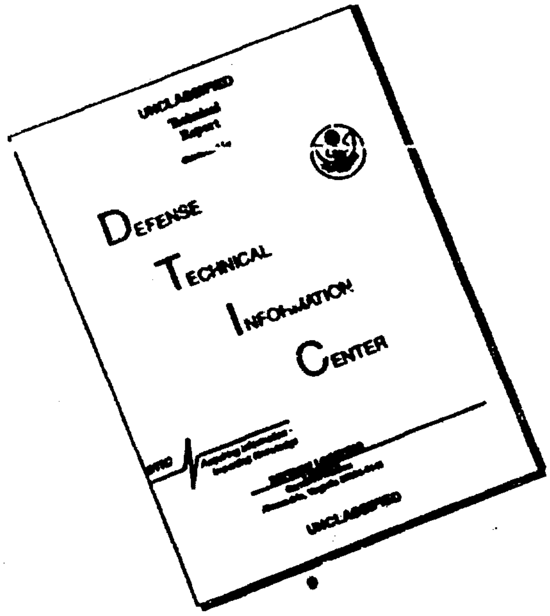
Thesis Advisor: Luqi

Approved for public release; distribution is unlimited.

91-12646

91 10 2 029

# DISCLAIMER NOTICE




**THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) | 5. MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a. NAME OF PERFORMING ORGANIZATION<br>Naval Postgraduate School<br>6b. OFFICE SYMBOL (If applicable)<br>37 | 7a. NAME OF MONITORING ORGANIZATION |
| 6c. ADDRESS (City, State, and ZIP Code)<br>Monterey, CA 93943 | 7b. ADDRESS (City, State, and ZIP Code) |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION<br>8b. OFFICE SYMBOL (If applicable) | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
| 8c. ADDRESS (City, State, and ZIP Code) | 10. SOURCE OF FUNDING NUMBERS<br>PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO. |

11. TITLE (Include Security Classification)
FUNCTIONAL SPECIFICATION FOR A GENERIC C3I WORKSTATION (U)

12. PERSONAL AUTHOR(S)
Steven E. Anderson

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Year, Month, Day) | 15. PAGE COUNT |
|---|---|---|---|
| Thesis | FROM 2/90 TO 9/90 | September 1990 | 271 |

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB-GROUP | software specification, hard real time software, embedded systems, generic C3I workstation, next generation computer resources |

19. ABSTRACT (Continue on reverse if necessary and identify by block number)


As fiscal impacts continue to make the U.S. Navy's budget increasinly austere, diligent efforts are being made to enhance the Navy's warfighting capability and streamline its operations. In this environment, it is necessary to efficiently produce computer systems that meet fleet requirements and do so at minimal cost. This thesis illustrates a methodology that integrates rapid prototyping with formal requirements analysis of a Generic C3I Workstation.

Traditional requirements analysis and software engineering principles tend to misplace emphasis on the completeness and the comprehensiveness of initial software modeling. Using formal specification tools and techniques, rapid prototyping provides prototype developers with constraints, while deliberately omitting up-front detailed descriptions. Precise constraints are refined through the evolution of requirements in the light of empirical performance and demonstrable functionality.


| | |
|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS | 21. ABSTRACT SECURITY CLASSIFICATION<br>UNCLASSIFIED |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL<br>Luqi | 22b. TELEPHONE (Include Area Code)<br>(408) 646-2735<br>22c. OFFICE SYMBOL<br>CS/Lq |

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—606-24.

UNCLASSIFIED

A network-capable software system for a Generic C3I Workstation, with embedded decision-support systems and robust message passing and processing, would offer functionality beyond that which is traditionally used in fleet operations. Through automation advancements, C3I information processing could be performed more quickly, efficiently, and accurately. In this thesis, an abstract model of a high-speed, networkable, generic C3I workstation is presented. This model defines the fundamental goals for the experimental modeling and development of a large, hard-real-time Ada software system for the U.S. Navy.

Approved for public release; distribution is unlimited.

# FUNCTIONAL SPECIFICATION
# FOR A
# GENERIC C3I WORKSTATION

by

Steven E. Anderson
B.A., Trinity College, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL

September 1990

Accesion For
NTIS CRA&I ✓
DTIC TAB
Unannounced
Justification
By
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

Author: Steven E. Anderson

Approved by: Luqi, Thesis Advisor

Timothy Shimeall, Second Reader

Robert B. McGhee, Chairman
Department of Computer Science

## TABLE OF CONTENTS

# ACKNOWLEDGEMENTS

This thesis is dedicated to my wife Anne, for letting me come out to California and continue my education. Thank you for all of the support, editorial services, technical assistance, UPS packages, mail, and late-night telephone calls.

I would like to thank the Naval Surface Warfare Center for providing me with the opportunity to attend the Naval Postgraduate School on Long Term Training. Special thanks is extended to my chain of command, J. Gary Ferrebee (G12), David S. Malyevac (G10), Oliver Paul Credle (G), and Thomas A. Clare (D).

I would like to express my appreciation to several people who were closely involved with this thesis: my advisors, Prof. Luqi and Prof. Shimeall, LCDR Jeffrey Schweiger for his editorial and technical assistance as well as supplying Appendix B, LTJG Cengiz Kesoglu and LTJG Vedat "Software Through Vedat" Coskun of the Turkish Navy for providing valuable feedback during the development of Appendices C and D, and LT John Daley for his encouragement, technical insights and editorial services. Discussions with many of the faculty, staff and students in the Computer Science Department, too numerous to mention individually, contributed towards my understanding of this work and helped me to finish this thesis in the given time frame.

# I. INTRODUCTION

> Past track records show that it now takes eight to ten years to develop, test, and deploy a new fleet system. This excruciating pace has been the rule for C2 systems. A delay of this magnitude puts the fleet at the peril of operating with C2 that is not by any means 'state-of-the-art.' The 'waterfall' design approach that begins with requirements debate and proceeds through a maze of software standards is simply too laborious and slow. The waterfall approach must be abandoned in favor of a more responsive C2 systems development. [Ref. 14: p 117]

A major portion of modern C3I systems is software. With the advent of the new generation of highly capable workstations that support common operating systems such as Unix™, emerging graphics standards and the increasing use of Ada for portability of applications software, it is software development that drives the costs of new C3I systems. Thus, it is important to concentrate on software engineering methodologies that decrease both development time and costs. The integration of formal requirements with rapid prototyping is such an approach.

## A. COMMAND, CONTROL, COMMUNICATIONS & INTELLIGENCE

The Department of Defense defines *command and control* as:

> The exercise of authority and direction by a properly designated commander over assigned forces in the accomplishment of the mission. Command and control functions are performed through an arrangement of personnel, equipment, communications, facilities, and procedures which are employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission. [Ref. 32: p 74]

The term *Command, Control, Communications and Intelligence (C3I)* is defined as "the collective activities of command and control specifically emphasizing the need for transfer of information between persons and places and the intensive role of intelligence in command and control."

The National Command Authorities and United States Congress dictate American foreign policy. The Department of Defense directives and planning documents (e.g., the U.S. Maritime Strategy) state, in general terms, where the U.S. Navy is expected to be and what it is expected to do. As day-to-day convolutions in world politics take place, the U.S. Navy serves as a major instrument for enforcing American foreign policy.

The U.S. Navy, like nearly every complex organization, maintains a layered management infrastructure. Smaller regions are managed by lower level managers. Larger regions are managed by middle level managers. On top, there are very few executive managers (i.e., commanders in chief). What differentiates these layers are (a) resources and (b) perspective (i.e., focus and/or area of interest).

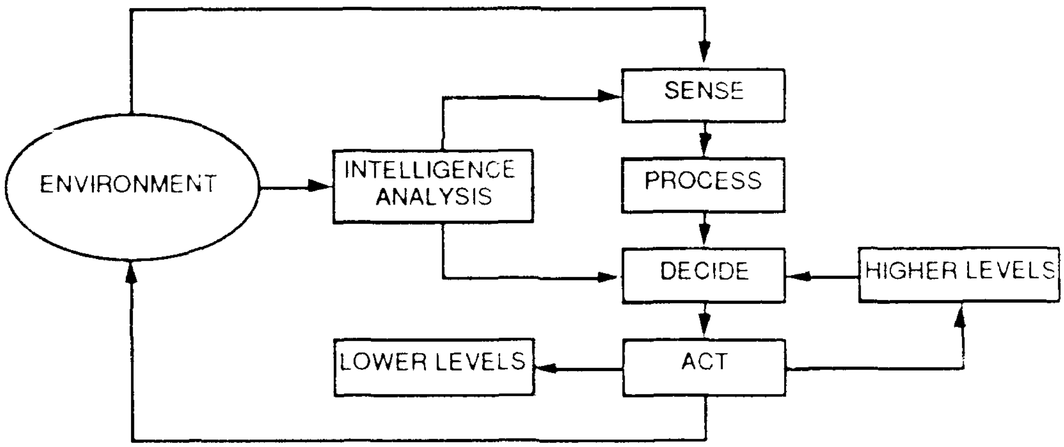



**Figure 1. Conceptual Combat Operations Process Model [Ref. 13: p 27]**

Figure 1 represents a idealized model of activities involved in combat operations. Regardless of where a commander may find himself withinin the chain of command, he will be performing similar evaluations: analyzing information concerning actions and events within their sphere of influence, determining what actions to take, responding to orders, and issuing orders.

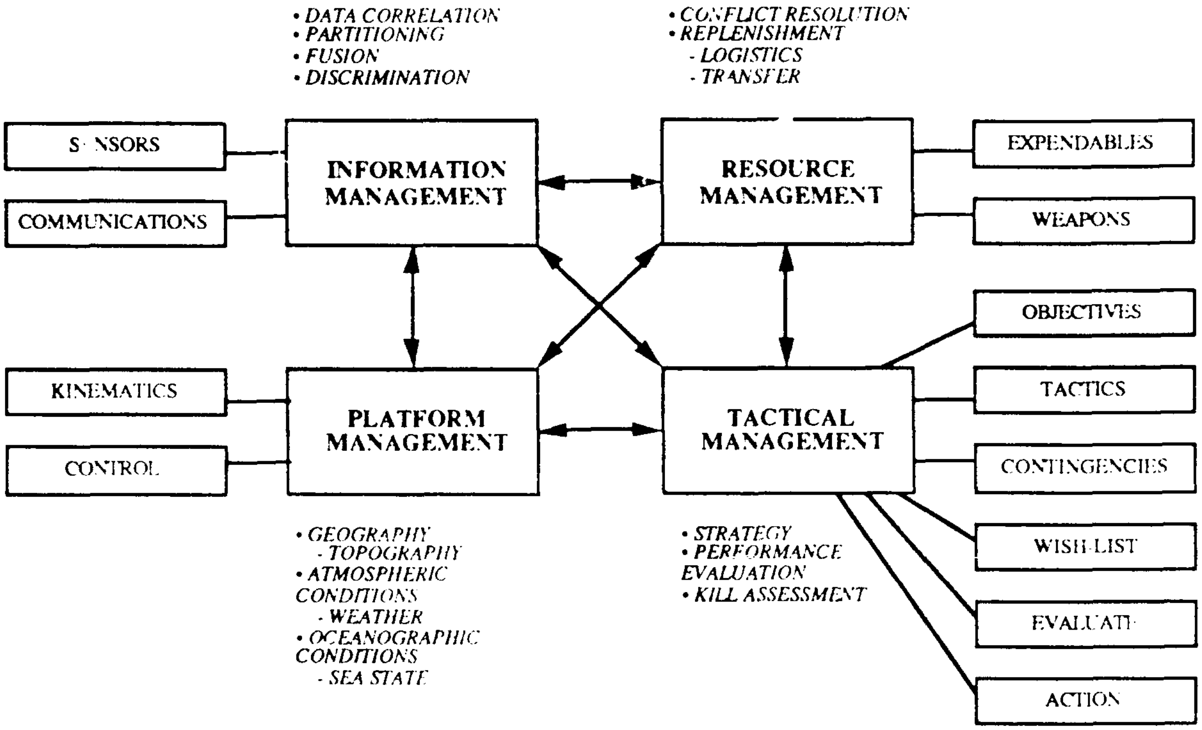



**Figure 2. A Generalization of C3I Management Functions**

Figure 2 partitions the fundamental activities identified in Figure 1 into general command, control, communications and intelligence functions. The four conceptual areas of management incorporated within C3I include: information management, resource management, platform management and tactical management. These four management areas will interact and overlap. In turn, each activity is concerned with managing a subset of C3I issues (as indicated by the smaller boxes).

### 1. Information Management

Technological advances such as radars and satellites permit the rapid collection of information from great distances. "[S]ystems and technologies have made C2 more difficult and time consuming for the commander -- pouring in data that clutter the decision making process instead of clearing it up." [Ref. 5: p 24]

Some of the major issues associated with integrating command and control information are: the volume of information to maintain, the differing types of information provided, the relative accuracies of reporting platform locations (gridlock), the relative accuracy of sensor systems, the timing delays associated with communications, and the lack of unique track identification.

Dissimilar source integration of information is difficult, and there are few tools and techniques presently available to permit the automation of this process. Further, the speeds at which vehicles travel, and the vast myriad of weapons that they are capable of employing underscore the risks that commanders take while evaluating incomplete or sporadic reporting information.

The Generic C3I Workstation abstract model deals with many of the issues involved in C3I information management. The Generic C3I Workstation is designed to accomodate a large number of tracks, integrate dissimilar source information and provide the commander with a timely tactical information.

### 2. Platform Management

Officers in command of ships, aircraft, and submarines must also control vehicular behavior. Perpetual concern is given to present position, relative locations of objects of interest, destinations, physical hazards, altitude/depth, flight envelopes, hostile weapons, countermeasures, cover and deception, cruising speed, docking, landing, damage control, reactor safety, inclement weather, terrain, visibility, etc.

At the current level of abstraction, the platform management issues associated with a patrol aircraft and a nuclear powered aircraft carrier differ so greatly as to make them difficult to incorporate into a common generic implementation. A Generic C3I Workstation implementation will need to interface with platform-specific platform management support tools as they become available.

### 3. Resource Management

Naval platforms may not be equipped with sufficient amounts of supplies to accomplish their intended missions. Ships, aircraft and submarines are capable of holding a finite amount of supplies, fuel, and/or weapons. The commander is a steward of his expendable resources. The commander constantly performs risk analyses to determine valid trade-offs between immediate use and potential needs.

The Generic C3I Workstation is designed to monitor weapons status. The Generic C3I Workstation could be expanded to monitor additional expendable resources as well. However, apart from weapons, the expendable resources associated with various C3I installations (e.g., ships, aircraft, submarines) differ sufficiently to make a generic implementation virtually impossible. A Generic C3I Workstation will need to interface with platform-specific resource management support tools as they become available.

### 4. Tactical Management

"Tactical and technological developments are so intertwined as to be inseparable. ... *To know tactics, you must know weapons.*" This is one of the Five Cornerstones of fleet tactics identified by Wayne Hughes. [Ref. 6: p 25]

Because today's ships, aircraft and submarines possess such a vast array of sensors, weapons, countermeasures, communications systems, etc., the decision processes associated with evaluating a tactical environment and determining what needs to be done, who should do it and when it should be done are complex indeed. While the U.S. Navy is experimenting with automated tactical weapons management [Ref. 6: p 189], tactics remain an art, mastered by practitioners. "C2 inevitably comes down to the decision maker, who must assess information, choose a course of action, give orders, and evaluate what happens. [Ref. 5: p 24]"

The Generic C3I Workstation is not designed to directly control weapons systems, rather it is designed to support the commander in controlling his assets. "Control is the act of executing decisions that have been made. Verbal, visual and electronic communications are the great instruments of control." [Ref. 6: p 189]

## B. GENERIC C3I WORKSTATION DEFINITION

### 1. Operational Context

Within the various fleets, task forces are grouped together and deployed as directed by designated authorities within the operational chain of command. According to the U.S. Maritime Strategy, there are five primary task force configurations: Aircraft Carrier Battle Force (CVBF), Battleship Battle Force (BBBF), Sea Lanes of Communication (SLOC) control force, Area Antisubmarine Warfare (AREA ASW) force, and Amphibious operations force (AMPHIB). (See Figure 3.) While the operational organizational structure within each of these task forces will vary greatly, depending upon the fleet, mission and resources, it is these battle group structures that the Generic C3I Workstation will support. Therefore, the Generic C3I Workstation must be adaptable to meet a variety of battle group structures.

An Aircraft Carrier Battle Group (CVBG) "has a battle group commander and warfare commanders in major areas: strike warfare, antiair warfare, electronic warfare, antisurface warfare and antisubmarine warfare." [Ref. 4: p 73] The battle group commander, often identified as the composite warfare commander (CWC) or officer in tactical command (OTC), will have authority over resource coordination and warfare mission area commanders. The warfare mission area commanders are in charge of the tactical control of assets (platforms and weapons). [Ref. 24: p 55] Further down the chain of command is the individual platform commander (i.e., officer in command of a ship, aircraft, or submarine). While the organizational structure within an aircraft is somewhat simple, a ship or submarine maintains a complex organizational structure within itself.

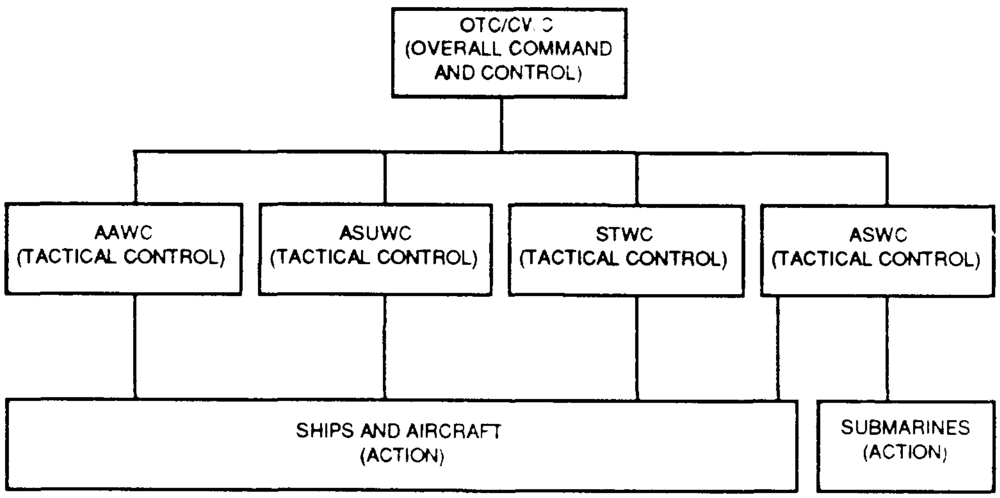



**Figure 3. Provisional Force Command Structure**

### 2. Workstation Description

The Generic C3I Workstation is designed to be installed on a wide variety platforms, in support of a Composite Warfare Commander (CWC) command and control architecture. The Generic C3I Workstation provides the CWC and his subordinate commanders and coordinators with a system that supports them in monitoring air, surface, subsurface, and power-projection (strike) tactical environments and aids in tactical decision making in those areas. The architecture provides for connectivity between naval platforms, shore-bases, and external forces and information sources, and enables the processing of tactical data from internal and external sources (where appropriate). [Ref. 18: p 10]

The Generic C3I Workstation makes use of an open system architecture enabling hardware modifications and upgrades without replacing the bulk of the system. Modular software design supports a variety of implementations while making use of reusable software components, such as the user interface, the track database management system, and message retrieval.

The Generic C3I Workstation is capable of displaying the current tactical situation within a geographical region in both graphical and textual forms. The graphical tactical display shows the most recent status of track information provided from own-ship platform sensor inputs, communications sources (including NTDS, OTCIXS, etc.), and manual inputs. The operator may set predefined and user-defined filters and precedence schemes to modify the system's behavior and restrict entries into the local database. The operator may also display a subset of the available track information based upon geographical regions, track type, or range.

An interactive communications dialogue capability to store communications messages provides the operator with the ability to read textual messages based on category, precedences, etc. The operator may use an intelligent message generator/text editor to call up message templates and interactively fill in the message.

The Generic C3I Workstation provides own-ship platform weapons status. Through automating the weapons status verification and availability process, weapons related information is readily available for either local battle damage assessment (BDA) purposes, or for situation report (SITREP) generation purposes. This supports the task force commander in determining how many assets are available to combat a known or potential threat.

## C. PARALLEL EFFORTS

"The U.S. Navy is shifting its perspective in command, control, communications, computers and intelligence -- reexamining functional and operational requirements in light of the breathtaking technological progress of the past two decades." [Ref. 8: p 58]

### 1. Existing Systems

The United States Navy does not have a C3I workstation that meets all of the following programmatic goals.

- a C3I system that maintains a consistent tactical picture across the battle group
- a C3I system that provides hard real-time information constraints upon a distributed database
- a C3I system that is fully capable of integrating tactical information from all available sources
- a C3I system that is generic, capable of being implemented on a wide variety of platforms
- a C3I system that supports the Navy's Next Generation Computer Resources (NGCR) effort, and is capable of running on a commercially available microprocessor
- a C3I system that is low-cost
- a C3I system that is non-proprietary
- a C3I system that is written in Ada

The number of existing command information systems within the fleet today is very small. The two systems most commonly mentioned in the open literature are the NTDS and the Aegis Combat System's Command and Decision subsystem.

#### *a. Naval Tactical Data System*

The Naval Tactical Data System (NTDS) serves as a representative C3I system currently in use by the fleet. The NTDS system "has been subject to a continuous process of updating since its introduction into service in the late 1950's." [Ref. 16: p 75] The system, viewed by today's standards, has numerous shortcomings. NTDS upgrades are attempting to replace "obsolete computers, displays and other equipment, and software modifications to reflect the integration of new sensors and weapon systems into the fleet." [Ref. 16: p 75]

Even though NTDS suffers from antiquated technology, it provides its users with "on-line collection, processing, storage, and presentation of information from sensors such as sonar, radar, optical, and aircraft or ship consorts, via data link." [Ref. 16: p 75] The Generic C3I Workstation must, as a minimum, provide modern NTDS-like functionality. By using current software development techniques and high-speed microprocessors, the Generic C3I Workstation will overcome most NTDS deficiencies experienced by the fleet today.

### *b. Aegis Command and Decision System*

The Aegis Combat System is a model of modern naval engineering. The Aegis Command and Decision (C&D) system already integrates information provided by sensors and communications systems. Aegis Baseline 5 provides for the inclusion of a Command and Control Processor (C2P) that integrates force level data from LINK-4A, LINK-11, LINK-16.

To a large extent, many of the functions that are envisioned for a Generic C3I Workstation are already embedded into the Aegis Combat System. The primary differences between the Generic C3I Workstation and Aegis are that the workstation is intended to provide a common communications interface, generic (non-platform specific) implementation, two way communications support, and functionality for battle force level command and control. The Generic C3I Workstation does not control weapons system or provide sensor coordination or cueing, as Aegis does (or will). Further, it is the intention of the Naval Postgraduate School that the Generic C3I Workstation will be written in the

Ada computer language and implemented on commercially available microprocessors in support of the Navy's Next Generation Computer Resources (NGCR) program.

### 2. Research and Development

The Navy's automation of the composite warfare commander's (CWC) C3I functions focuses on two major systems, known as Navy command and control system, afloat (NCCS-A) and the advanced combat direction system (ACDS).

NCCS-A will serve as the principle system supporting the CWC/officer-in-tactical command (OTC). NCCS-A improves the tactical flag command center (TFCC)/flag data display system (FDDS) and integrates TFCC/FDDS with the afloat correlation system (ACS) and the electronic warfare coordinator's module (EWCM). NCCS-A will support the CWC/OTC in planning and executing battle force/battle group operations, provide dynamic tactical situation displays, provide for functional interaction between tactical warfare commanders, and provide connectivity to numbered fleet commanders and fleet commanders-in-chief. ACDS will replace the current Navy tactical data system (NTDS) and can support initiatives such as the anti-submarine warfare commander module (ASWCM) for direct support of the individual tactical warfare commanders. NCCS-A will be a network making use of open system architecture to interface various current and future C3I systems, including ACDS, to support the CWC. In the near term, easily programmable commercial, off-the-shelf desktop computer hardware and software have automated previously manhour-intensive C3I functions. [Ref. 31: p 29 - 30]

#### *a. Navy Command and Control System, Afloat*

The Navy Command and Control System, Afloat (NCCS-A) is managed by the Space and Naval Warfare Systems Command, PMW-162. NCCS-A is composed of several in service programs, including: the Tactical Flag Command Center (TFCC), the Joint Operational Tactical System (JOTS), the Prototype Ocean Surveillance Terminal, the TFCC Information Management System, the Interim Command and Control System, the Naval Intelligence Processing System, the Fleet Imagery Support Terminal, and secure Closed Circuit Television. [Ref. 36: p 1]

Most of these systems are in their initial development phase. The Generic C3I Workstation, as presented, maintains significant overlaps with many of these systems. For instance, "TFCC provides tactical displays, integrated information management systems and accessibility to tactical communications. TFCC will provide an accurate, redundant, survivable, distributed and consistent tactical picture." [Ref. 36: p 3] These are also the goals of the Generic C3I Workstation. "JOTS provides a near real-time battle management system for tactical decision support, including: tactical decision aids, message processing, tactical data management, tactical overlays, environmental predictions, and general planning aids." [Ref. 36: p 5] The Generic C3I Workstation's tactical displays and message processing functions would be very comparable with JOTS. The Generic C3I Workstation would provide two-way message handling, provide provide support to multiple networks. The Generic C3I Workstation does not provide tactical decision aids, prediction and planning functions, however its modular design could easily support these extensions.

The Generic C3I Workstation and the NCCS-A systems address similar requirements but are parallel efforts. NCCS-A is a formal multi-million dollar Navy acquisition program that provides operational commanders with effective C3I tools and equipment. The Generic C3I Workstation is a small-scale research effort conducted at the Naval Postgraduate School and sponsored by the sponsor of NCCS-A to evaluate the rapid prototyping methodology in satisfying similar requirements at lower cost in the future. The Generic C3I Workstation will make use of standards and requirements provided by the NCCS-A whenever possible (cf. the *Software Requirements Specification for the NCCS-A Workstation Executive, Volume 1: Man-Machine Interface*). Through the rapid prototyping of the Generic C3I Workstation, valuable contributions could be made to the development of real-time software development, track database design, Ada coding and system performance constraints, and thus positively impact NCCS-A efforts.

### b. *Advanced Combat Direction System*

The currently deployed combat direction systems (CDS) may be characterized as little more than a "display system" offering naval officers little real-time tactical support. [Ref. 15: p 118] The Advanced Combat Direction System (ACDS) brings new hardware and software (software technology) to the fleet. Embedded decision support will provide ACDS users not only with an accurate representation of the immediate tactical situation but will assist the Tactical Action Officer and Combat Information Center personnel in responding quickly and decisively to real (or potential) threats. Thus the ACDS (in Block-1) will bring "carriers, non-Aegis cruisers, and potentially many other ships up to and beyond the tactical capability baseline established with Aegis." [Ref. 15: p 119]

The ACDS differs in many ways from the Generic C3I Workstation. The ACDS is a shipboard system, for use by ship personnel in assessing the local tactical situation and providing platform specific responses to threats. The Generic C3I Workstation supports battle group operations by providing an accurate picture of the battle group tactical situation. The Generic C3I Workstation also stresses two way communications support between battle group commanders, while CDS systems primarily serve in a passive (receive only) role, and supports a smaller tactical region.

### c. *Next Generation Computer Resources*

The Naval Research Advisory Committee on Next Generation Computer Resources (NGCR) has been tasked to assess reasons why the Navy's computer technology lags so far behind the current state-of-the art, as well as to provide guidelines for cost effectively infusing newer technology into the fleet.

> The objective of the NGCR program is to support improved fleet operational readiness by providing a family of state-of-the-practice computer resources using a

flexible open architecture that will significantly reduce the costs, complexity and schedule requirements associated with system integration and that can be easily adapted to changing system requirements. The objective will be accomplished through the definition of commercially based non-proprietary hardware and software interface standards and protocols that will be applicable to all [Mission Critical Computer Resources] for new systems. Selected, critical standards will be supplemented with laboratory test modelling to validate their correctness and with the establishment of a conformance process to certify vendor hardware and software compliance with the published standards. These standards, based on an open systems architecture, will be jointly defined by industry and the Navy to take maximum advantage of ongoing commercial trends and standardization activities. [Ref. 35: p 1]

NGCR effort recognizes that valuable encapsulated software and ruggedized versions of hardware are commercially available. Instead of developing expensive one of a kind systems, the goal now is to make as much use as possible of commercial standards and "off the shelf" technology. [Ref. 33: p 2]

#### d. *Naval Postgraduate School*

Ongoing research in support of C3I functions is being conducted at the Naval Postgraduate School. A team of faculty and students from the Computer Science Department is working on a Combat Direction System in Ada and portable to shipboard computer systems. Continued progress is being made in the automatic generation of the man-machine interface (MMI), object-oriented database management systems, software prototyping of hard real-time systems, Ada coding of the CDS, and parallel processing.

All of these research areas are viewed to be directly applicable to the development and implementation of a Generic C3I Workstation prototype. Operational software such as that provided by a Generic C3I Workstation must eventually run on shipboard computer systems. A modern full-feature MMI for tactical displays and message generation is a key feature of the Generic C3I Workstation. Further, novel approaches to solving distributed database problems, as well as database systems that support hard-real-time constraints are of particular interest to the Generic C3I Workstation.

## D. RAPID PROTOTYPING

### 1. Prototyping Methodology

As previously indicated, traditional software methodologies, such as the waterfall model, are often too slow and too costly to be appropriate for C3I systems development. The waterfall model, as presented by Royce [Ref. 17: pp 1 - 9], imposes a linear abstraction onto the iterative process of system software development. (See Figure 4.)

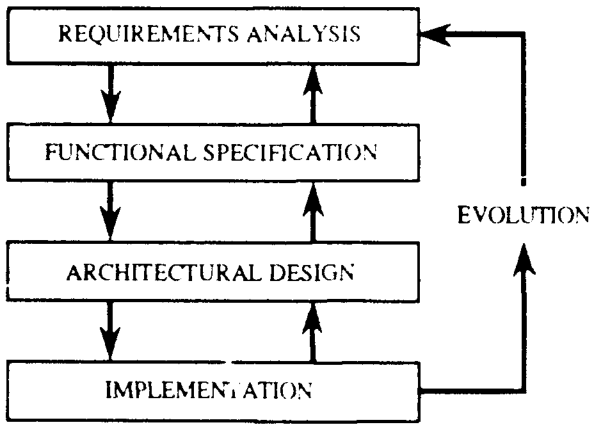



**Figure 4. Software Engineering Development Activities [Ref. 1: p 7]**

Early proponents of the waterfall model supported the notion that system requirements must be determined as completely as possible prior to the design, implementation and testing of the proposed system. Unfortunately, when major modifications to system requirements are made late in the development process, the time, effort, and money required to retrofit changes becomes significantly higher than if they were made up-front. Hence, requirements analysis is perhaps the most critical stage of the software development process, since this is when the system is defined. [Ref. 25: p 3]

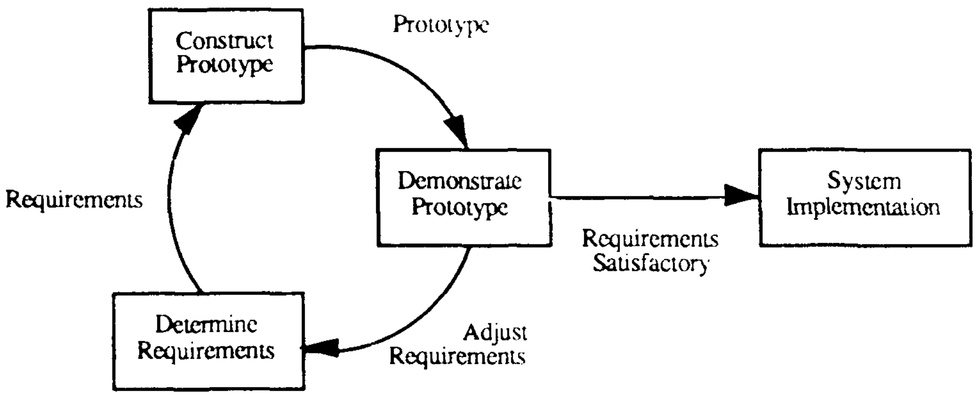



**Figure 5. Prototyping Life Cycle [Ref. 10: p 30]**

Requirements are often incomplete, incorrect or inapplicable due to poor communication between the users and the software developers. The software developer's lack of understanding of the application problem domain further undermines the requirements specification process, and the quality of the end-product.

Prototyping was developed as an alternative to the waterfall life cycle model for software development. An iterative cycle is entered with the users and the development team. Requirements are made, a prototype is constructed, and demonstrated for the user. The user may then clarify or modify the development team's understanding of the requirements, and the cycle is repeated. (See Figure 5.)

### 2. Computer-Aided Prototyping System

By automating as much of the development process as possible, iterative prototype construction and requirement feedback may be accomplished more quickly. [Ref. 25: p 5] An automated support environment is essential for constructing prototypes *rapidly*. [Ref. 10: p 29] Thus the term *rapid prototyping* refers to computer-aided prototyping development systems.

At the Naval Postgraduate School, the Computer-Aided Prototyping System (CAPS) is an experimental system to support rapid prototype development. CAPS requires a developer to first describe a new system concept in the Prototype System Description Language (PSDL). [Ref. 12: pp 66 - 69] "This description specifies the system as a network of components. Each component is described by name, parameters, results and timing behavior, possibly augmented by a series of axioms denoting the effect of the component, by free-form comments and by a description of the implementation of the component. After the system is described in PSDL, CAPS [will match] each component against a library of [software] modules, looking for combination of one or more modules that implement that component. Any components that have no matching modules are submitted for module development. When all components have an associated implementation, an executable prototype is generated." [Ref. 20: pp 4 - 5]

The requirements provided by this thesis will be serve as a baseline in the rapid prototype of the Generic C3I Workstation. Immediate follow-on efforts will take the Yourdon functional decomposition of the Generic C3I Workstation (see Appendices C and D) and create a set of PSDL specifications. These PSDL specifications will subsequently be fed into CAPS, and yield executable Ada code.

### 3. Requirements for Prototyping Efforts

"A prototype system is not a production system. The purpose of a prototype is to provide answers to questions about the requirements and the properties of the proposed system. The prototype must include only the aspects of system behavior relevant to answering these questions. It does not have to be complete, reliable or efficient." [Ref. 10: p 31]

The Generic C3I Workstation maintains incomplete requirements specifications for a number of deliberate reasons (and some not so deliberate reasons). Wherever bonafide requirements constraints could be cited, they are. However, the following three considerations must also be taken into account.

#### a. *Classified Data*

Military command, control, communications and intelligence is, by its very nature, secretive. Most performance parameters and message formats associated with C3I systems are classified. Since the Generic C3I Workstation is being developed in an unclassified forum, deliberate care has been exercised to avoid the use of classified values. Representative sanitized values have been substituted for actual parameters wherever possible.

#### b. *Anticipated Prototype Reviews*

A great deal of the user interface is deliberately not being specified up front. A very straight forward interface will be adopted for initial review purposes. As the user refines their understanding of the Generic C3I Workstation interface, the input/output formats will evolve.

A distinct problem with the Generic C3I Workstation effort is that nobody has ever built one before. While "best guess" values may be adopted, many values will be left to the prototyping team to determine. Occasional values have been chosen arbitrarily to provide a baseline set of values for the prototyping effort. Once empirical values are determined, they will be substituted in later iterations of the prototyping cycle.

#### c. *Project Limitations*

As has been stated earlier, a prototype is not a production system. The scope of this effort at the Naval Postgraduate School has been limited to one year. While the development of a production system is highly feasible, deliberate choices have been made to limit the scope of the prototyping effort in order to make it tractable.

Tremendous effort could be exerted to ascertain specific values associated with likely hardware interfaces (such as LINK-11 data terminal sets, radars, sonars, weapons systems, etc.). However in the interest of the economy of effort, generalizations and assumptions are made concerning these systems for the prototyping effort. Certainly for a production system, such interfaces constraints would be of paramount importance.

## E. RESEARCH OBJECTIVES

The Generic C3I Workstation is a new research effort. There does not exist any documentation on the desired system. The Naval Tactical Interoperability Support Activity (NTISA), San Diego, CA, is in control of setting U.S. Navy communications standards. The Naval Ocean Systems Center (NOSC), San Diego, CA, is the Navy's lead lab in support of fleet C3I activities. Documents and interviews provided by these activities have provided the background setting for this thesis. The Next Generation Computer Resources (NGCR) Program has also significantly shaped the outcome of this research effort.

This thesis is the first of a group of related theses. The result of this research is to provide an initial requirements statement and functional specification for a Generic C3I Workstation, which will support a prototype research effort. The objective of this work is to provide the following:

(1) A model of the prototype system's environment

(2) A description of the initial goals of the system and functions it must perform

(3) Performance constraints on the prototype system

(4) Constraints on the environment and implementation of the prototype system

(5) Recommended design approaches based on available technology and experience with existing systems.

Software design and initial software modeling are perhaps the most difficult aspect of the software engineering process. This thesis provides a general software design, requirements specification, functional specification, and abstract model for a Generic C3I Workstation. The analysis provided by this research will be used as the foundation for a rapid prototype development effort of a Generic C3I Workstation, making full use of modern software engineering principles.

## F. THESIS ORGANIZATION

Chapter II provides the initial statement of requirements and constraints for a Generic C3I Workstation. Chapter III uses the Yourdon approach to structural analysis and design of the proposed Generic C3I Workstation. Chapter IV provides a top-level system specification for Generic C3I Workstation networking and introduces work being done by LCDR Jeffrey Schweiger on "Generation of a Deadlock Determination Tool for the Spec Formal Specification Language". Chapter V describes an overview of implementation considerations, suggests a list of operational system designs, and introduces follow-on prototyping efforts by LTJG Cengiz Kesoglu and LTJG Vedat Coskun entitled "Software Prototypes of C3I Stations." Chapter VI provides a summary and conclusion, as well as a listing of suggested areas of research.

Appendix A provides a glossary of C3I terms used in this thesis. Appendix B provides an initial functional specification for a Generic C3I Workstation written in the SPEC specification language and developed by LCDR Jeffrey M. Schweiger. Appendix C contains a set of data flow diagrams that comprise a functional decomposition of the Generic C3I Workstation. Appendix D is a preliminary set of process specifications which correlate with Appendix C. Appendix E is the data dictionary for Appendix C. Appendices C, D, and E represent work original to this thesis effort. Appendix F provides a list of acronyms and abbreviations used in this thesis.

# II. WORKSTATION GOALS AND CONSTRAINTS

## A. INITIAL GOALS AND REQUIREMENTS

To be an effective support tool, the Generic C3I Workstation must provide battle group commanders with the ability generate, transmit, receive and process tactical information. In support of a commander, the Generic C3I Workstation must provide accurate, complete (non-redundant, non-extraneous), and timely information that may be used to make prompt and effective decisions. The sort of information a commander needs includes: (a) kinematic information (What tracks are out there? Where they are? How are they moving?), (b) intelligence information (Who is out there? What are they doing? What is their intent or objective?) and (c) operational information as relayed by other commanders (What are the current action orders? What are our current mission goals and objectives? Are there any local constraints or considerations that need to be taken into account? Are there logistics considerations that need to be resolved?)

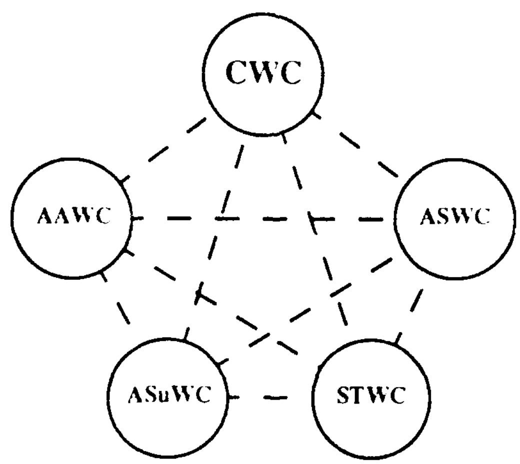



Figure 6. Battle Group Level Connectivity [Ref. 37: p 5]

This effort develops the description of a C3I workstation that supports the CWC, warfare mission area commanders, and force coordinators in conducting battle group C3I functions (see Figure 6). The Generic C3I Workstation must provide connectivity between U.S. Naval surface ships, aircraft, submarines and land bases, by providing the real-time ability to receive, process, and transmit tactical data from many interfaces in support of a CWC Command and Control Architecture. The following goals apply to the Generic C3I Workstation prototype effort.

### 1. C3I Functionality

G.1. The Generic C3I Workstation must provide battle group commanders with the ability generate, transmit, receive and process tactical information.

G.1.1. The Generic C3I Workstation must be able to acquire tactical data from multiple sources.

G.1.1.1. The Generic C3I Workstation must be able to receive and display textual communications messages.

G.1.1.2. The Generic C3I Workstation must be able to receive communications messages and extract relevant information concerning the position, constituency and kinematic behavior of a set of tracks.

G.1.1.3. The Generic C3I Workstation must be able to receive and maintain information from platform sensors that provide the position, constituency and kinematic behavior of a set of tracks.

G.1.1.4. The Generic C3I Workstation must provide the user with information concerning relevant platform weapon status information.

An abstraction of U.S. Navy command and control messages reveals two fundamental categories: data messages and text messages (see Figure 7). While these classes are not mutually exclusive, data messages represent machine-to-machine transactions which may be unintelligible to anyone other than the system developer (e.g., NTDS track data, network protocol messages, positioning information, etc.). Text messages refer to communications information that is system independent and is capable of being displayed in character format and read by a human operator. Such a characterization of naval command and control messages is both link and transmission protocol independent.

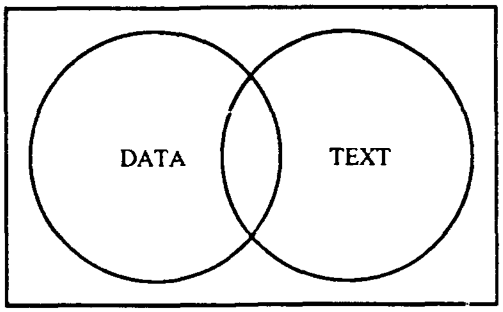



**Figure 7. Generalization of Naval Command and Control Messages**

Based upon this abstraction, the Generic C3I Workstation must be able to process and display textual messages and graphical data. Through an open systems architecture and modular design, the Generic C3I Workstation is capable of quickly and accurately transmitting, receiving, or displaying tactical information passed via textual or data formats. The specific algorithms required to parse, interpret and decode a particular format must be implemented within the Generic C3I Workstation.

G.1.2. The Generic C3I Workstation must be able to store, process and update tactical information from multiple sources in real time.

G.1.2.1. The Generic C3I Workstation must be able to parse incoming communications messages and extract track/contact information in real time.

G.1.2.2. The Generic C3I Workstation must be able to parse incoming sensor-related messages and extract track/contact information in real time.

G.1.2.3. The Generic C3I Workstation must provide a track-database system capable of accessing and updating track information in real time.

The Generic C3I Workstation will receive tactical information from platform sensors and communications links (see Figure 8) and synthesize tactical information into a coherent picture. Information synthesis is a very difficult task. Tactical information will come from multiple sources in different formats, at varying times, with incomplete, inconsistent, or contradictory information, and with varying degrees of accuracy.

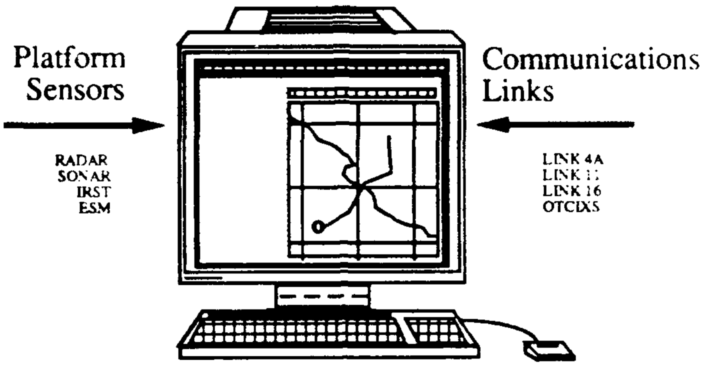



**Figure 8. Generic C3I Workstation Tactical Input Sources**

"You hear a lot in the C3I world about filtering and a lot about fusion. I think one of the biggest problems about fusion of data is inattention to a simple rule: if you want to bring diverse information together and have it make sense, you must have some understanding of what is going on." [Ref. 7: p 9]

Figure 9 indicates the potential time lag associated with information provided from communications links compared with platform sensors. In general , information updates provided over communications links will be less frequent and delayed longer than information updates from platform sensors ($\Delta_1$ versus $\Delta_2$). When interacting with information from multiple sources, the most recent information should be displayed, unless specified otherwise by the user. For instance, in a distributed database environment, a Force Over-the-horizon Track Coordinator may be relegated to maintain a centralized (or official) set of tracks, and periodically transmit this track information to network participants. While FOTC information may be a few seconds old, it still serves as the sanctioned battle group track database.

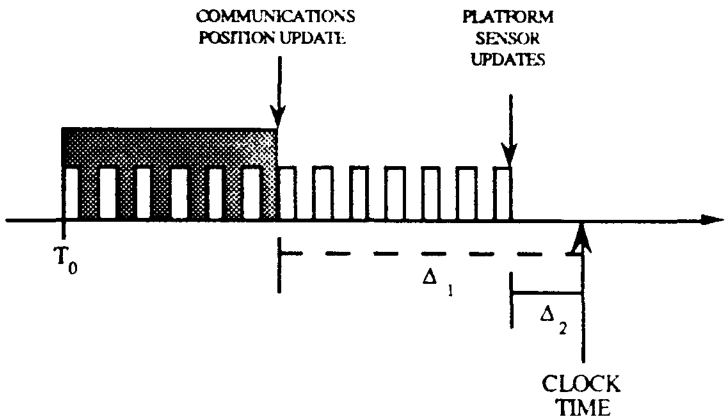



**Figure 9. Timing Considerations**

The Generic C3I Workstation prototype is a tool that displays the most recent tactical information provided by communications or sensors. An operational version of a C3I system will need to display tactical information based upon a complex definition of *track quality*, rather than based simply on timeliness. (N.B., "Track quality" will be defined in large measure by sensor system performance and has be omitted from this thesis due to its classified nature.)

The continuity of track reporting, or redundancy of track reports will be resolved by an expert system embedded within the Generic C3I Workstation. For prototyping purposes, track information will be included somewhat indiscriminately into the track database. The Track Database Monitor (expert system) will periodically scan the track-database in order to match, merge, and correlate track information based upon simple extrapolation algorithms (i.e., dead-reckoning).

G.1.3. The Generic C3I Workstation must provide the user with a form-based syntax-directed editor for rapidly generating naval communications messages.

G.1.3.1. The Generic C3I Workstation must provide the user with a modern user interface that makes use of a graphical interface, windows/menus, indirect pointing devices (mouse/track-ball), etc. for the purposes of assisting in the generation of communications messages.

G.1.3.2. The syntax-directed editor must supply the user with the most recent values and information available for inserting into designated template slots.

G.1.3.3. The Generic C3I Workstation must be able to automatically provide updated message information for the periodic transmission of reporting messages.

C3I information processing is a manually intensive task. If some of these C3I functions could be automated, then they could be performed more quickly, efficiently and accurately. The Generic C3I Workstation makes use of modern user interfaces, form-based syntax-directed editors and embedded decision support systems to enable the user to quickly generate and forward messages.

The Generic C3I Workstation includes a syntax-directed text editor that makes use of modern interface tools and techniques. User-generated messages employ message templates and require the user to fill in minimal amounts of necessary information while the system automatically provides values and information to be inserted into designated slots. Syntax-directed editors may provide initial constraint violation checks.

### 2. Information Update and Display

G.2. The Generic C3I Workstation must be able to provide complete and robust displays of real-time tactical data.

G.2.1. The Generic C3I Workstation must provide the user with the ability to customize and partition information.

G.2.2. The Generic C3I Workstation must provide the user the ability to adjust information-update rates wherever practicable.

G.2.3. The Generic C3I Workstation must provide the user with a multiple windowing environment.

G.2.3.1. The Generic C3I Workstation must provide the user the ability to limit the number of display elements.

G.2.3.2. The Generic C3I Workstation must clearly identify that the user is viewing a subset of tracks from the track-database.

G.2.4. The Generic C3I Workstation must provide the user to vary his geographic region of interest.

G.2.5. The Generic C3I Workstation must provide the user the ability to adjust a variety of parameters affecting tracks of interest, and behavior-threshold characteristics.

Through user-defined parameters that control the behavior of a series of filters and queueing precedences, a Generic C3I Workstation installation may control the quantity, quality and variety of information to be processed. Thus, if a particular battle group Antiair Warfare Commander (AAWC) were only interested in air track information, he could either filter out any unwanted tracks from entering his system in the first place, or he could permit his system to maintain a larger set of track data while he selectively displays only those tracks of interest. Relative precedences could be given to messages, such that particular types of messages, or messages received from particular senders would be processed more quickly than others within particular instance of the Generic C3I Workstation. In such a manner, the user may define and redefine the environmental perspective of his particular workstation.

### 3. Communications Networking

G.3. The Generic C3I Workstation must be able to participate as an active element in communications networks.

G.3.1. The Generic C3I Workstation must be able to accurately receive a wide variety of network communications.

G.3.2. The Generic C3I Workstation must be able to transmit and forward communications messages over a variety of communications networks.

G.3.3. The Generic C3I Workstation must not send classified information over a communications network that exceeds the network's classification.

By enabling a C3I workstation to serve as a gateway between communications links with similar functionality, a commander is provided alternative methods for receiving information from multiple sources, as well as alternate communication paths for transmitting messages.

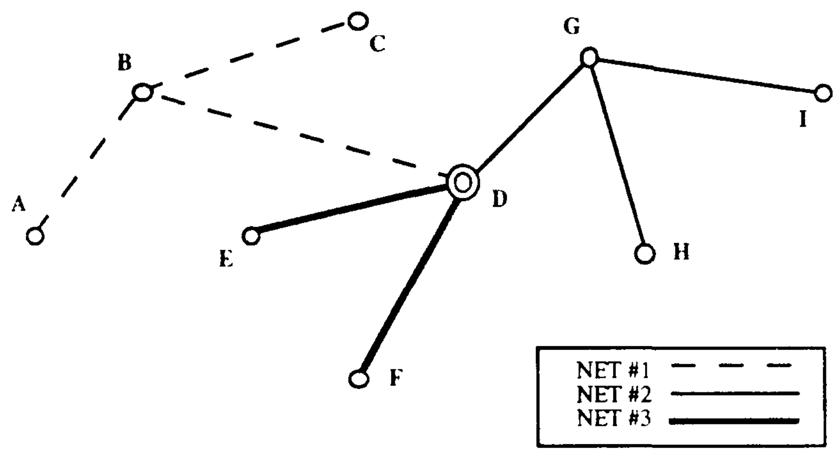



**Figure 10. Communications Network**

Figure 10 illustrates the possible connectivity enhancements due to an open system architecture supporting communication gateways. Here, Platform D serves as a relay station capable of passing information between elements in Net #1, with any of the elements in Net #2 or #3. Platform D may collect data from any of the three nets.

Implicitly a communications network gateway must provide the ability to translate from one message format to another, resolving inconsistent information parameters. This represents no small task, since information provided by one link may have an entirely different word size, degree of accuracy, and message organization than another. For targeting purposes, the accuracy of a track should be explicitly maintained by the Generic C3I Workstation track-database, rather than being implicitly determined by a communications link. Efforts must be made to minimize the loss of accuracy due to message translation.

Messages, and message information capable of being mapped from one communications message format to another could be automatically converted by the C3I workstation for inter-networking purposes. While additional research in message format conversion needs to be pursued, Figure 11 indicates a notional translation mapping for one communications format standard (such as, the Over-The-Horizon Targeting (OTH-T) GOLD Reporting Format used with the Officer in Tactical Command Information Exchange System (OTCIXS), to another format (such as, the United States Message Text Format (USMTF) reporting format used in conjunction with the Joint Interoperability of Tactical Command and Control Systems (JINTACCS)).

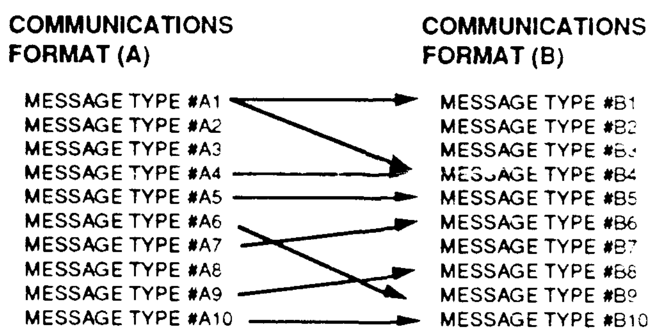



**Figure 11. Notional Message Translation Mapping**

The Generic C3I Workstation will contain a communications network database, that will identify the communications links connecting the sender to known addressees. A network database offers to alleviate considerable communications network information currently retained by human operators. Hence, after a user has created a communications message and identified the addressee, the system will automatically send the message via an appropriate communications link. Regardless of whether the addressee is located aboard the same platform as the Generic C3I Workstation, or must be reached via satellite communications links, the system will correctly forward the message.

Naval communications systems often have been designed for specific purposes and are not easily expandable or adaptable. Some communications systems specialize in unique or idiosyncratic data transmissions that are not of any particular interest to command and control systems. Hence, certain message types may not be able to be mapped to other formats. Conversely, it is possible that one particular message type may correlate to multiple message types in other formats. Specific mappings from one one message format standard to another will be classified.

### 4. Future Goals for an Operational Generic C3I Workstation

Beyond the scope of the prototyping effort, many additional goals can be envisioned for operational versions of the Generic C3I Workstation. These goals include the following:

G.4. The Generic C3I Workstation must be able to interface with other shipboard and remote systems.

G.4.1. The Generic C3I Workstation must be able to provide real-time track information to other systems.

G.4.1.1. The Generic C3I Workstation must be able to provide real-time targeting-quality track information to weapon systems.

G.4.1.2. The Generic C3I Workstation must be able to provide real-time cueing information to platform sensors, for improved sensor coordination.

G.4.1.3. The Generic C3I Workstation must be able to provide track information to tactical decision aid tools.

G.4.1.4. The Generic C3I Workstation must be able to include platform sensors information within communication messages.

G.4.2. The Generic C3I Workstation must be able to receive and display additional external information for tactical situation displays.

G.4.2.1. The Generic C3I Workstation must be able to receive and display weather information.

G.4.2.2. The Generic C3I Workstation must be able to receive and display information contained in the Naval Warfare Tactical Database.

G.4.2.2.1. The Generic C3I Workstation must be able to receive and display weapon engagement ranges.

G.4.2.2.2. The Generic C3I Workstation must be able to receive and display performance data for a given track.

G.4.2.2.3. The Generic C3I Workstation must be able to receive and display the information concerning the location and facilities of military bases and strategic targets.

G.4.2.3. The Generic C3I Workstation must be able to receive and display radar coverage zones.

G.4.3. The Generic C3I Workstation must be able to interface with instantiation-specific platform management tools, as they become available.

G.4.3.1. The Generic C3I Workstation must be able to provide platform management tools with real-time information concerning own-ship.

G.4.3.2. The Generic C3I Workstation must be able to receive and display navigation plans and routes.

G.4.3.3. The Generic C3I Workstation must be able to receive navigation plans and routing information and include this information into communications messages.

G.4.4. The Generic C3I Workstation must be able to interface with instantiation-specific resource management tools, as they become available.

G.4.4.1. The Generic C3I Workstation must be able to provide resource management tools with real-time information concerning own-ship resource statuses (i.e., weapons status).

G.4.4.2. The Generic C3I Workstation must be able to receive and display additional own-ship resource status reports (i.e., battle damage, flooding, reactor status, fire, casualties, etc.).

G.4.4.3. The Generic C3I Workstation must be able to receive own-ship resource status reports and include this information into communications messages.

## B. PROTOTYPE CONSTRAINTS

The prototype Generic C3I Workstation is not a production system. The prototyping effort associated with the Generic C3I Workstation serves as a basis for research into the study of prototyping design tools for developing requirements specifications and implementation tools for rapidly constructing prototypes (such as CAPS).

There are a number of constraints that apply to this prototyping effort. The following sections address the key constraints affecting the Generic C3I Workstation effort here at the Naval Postgraduate School.

### 1. Resource Constraints

The constraints affecting the time, effort and money invested into the Generic C3I Workstation prototyping effort are managerial in nature. While the development constraints for the Generic C3I Workstation are largely unlimited, the requirements analysis (provided herein) is scheduled to be completed by the end of FY90. The deadlines for producing the architectural design and prototype development phases are uncertain.

### 2. Implementation Constraints

Chapter V provides a more detailed prototyping implementation model, and description of initial prototyping efforts. The following discussion provides high-level constraints that affect the prototyping development efforts here at the Naval Postgraduate School.

#### *a. Unclassified Environment*

The Generic C3I Workstation prototype shall remain unclassified. Only unclassified message formats shall be modeled within the system prototype. Only unclassified information shall be maintained by the system prototyping. Prototype software shall not include any classified algorithms, data or protocols. The prototyping hardware may include security features such as removable memory, yet this is not required. The Generic C3I Workstation prototype shall be developed at the Naval Postgraduate School, using non-secure computer resources, and may be accessible by personnel who do not maintain Department of Defense security clearances.

#### *b. NGCR Hardware and Software*

The Generic C3I Workstation shall implement the basic features of a command and control system using commercially available computer hardware and software resources in keeping with the (proposed) Operational Requirement for Next Generation Computer Resources (NGCR). This requirement advocates the "prototyping [of] computer resources for specific major weapons systems using ruggedized commercial equipment, commercial software tools and applications, Ada, and incorporating widely used commercial standards." [Ref. 33: p 53]

In accordance with Department of Defense directives, Ada shall be used as the implementation language for the Generic C3I Workstation prototype.

### 3. Performance Constraints

Most of the hard-real-time constraints placed upon the Generic C3I Workstation are mandated by the specific external equipment interfaced with a specific C3I workstation instantiation. The United States Navy maintains interface design specifications (IDSs) that identify engineering-level format and protocol information required to interconnect operational military systems. When designing external system interfaces, development team members are encouraged to make use of existing IDSs whenever possible, and thoroughly familiarize themselves with equipment interfaces for which no IDS exists.

External system interfaces are of two types: synchronous and asynchronous. Systems that provide information on a regular or periodic basis are said to be *synchronous*.

Systems that provide information updates at unpredictable intervals are said to be *asynchronous.*

The Generic C3I Workstation prototype shall simulate synchronous and asynchronous external interfaces in keeping with the external systems identified in the behavioral model (see Chapter III).

*a. Synchronous Systems*

(1) Navigation System. Most navigation systems will provide position update information at regular time intervals. For prototyping purposes, it is assumed that the Navigation System will transmit a message containing the platform's course, latitude, longitude, velocity, altitude/depth, and Greenwich Mean Time (GMT) approximately once every second. [Ref. 3: pp 33 - 35]

(2) Periodic Communications Updates. While communications systems are asynchronous, they may often transmit messages periodically. Data update rates for specific communications messages and systems are classified.

There will be a direct correlation between the number of tracks in a given region and message processing delays. The fewer the targets, the less information needs to be processed. The more targets, the more information needs to be processed. Hence, in a target enriched environment there will be greater computational demands placed upon a C3I system.

Provisionally, the Generic C3I Workstation should be capable of retrieving data from up to 1000 tracks in less than one second[1] . From the time a track data

[1] The Center for Naval Analysis' AAW Masterplan and Sea War '85 "concluded that sensor platforms eventually must be capable of simultaneously tracking up to 3,000 objects -- friendly and hostile -- aircraft, surface ships and submarines, as well as commercial aircraft and shipping." [Ref. Signal, Feb 1990: p 61]

message is received by the system and its contents are entered into the track database should be less than two seconds.

(3) Periodic Sweep Sensor Systems. Many sensor systems, including surface search radars and infrared search and target designation systems, are mounted on a rotating platform that spin at a constant rate. Thus after every 360° revolution of the antenna (or sensing device) new contact information will be available over the full coverage range. Nominally, periodic sweep sensor systems rotate once every five to ten seconds. It is assumed that all track data is updated within this time frame.

| | | |
|---|---|---|
| AN/SAR-8 | (2 sec update rate) | [Ref. 3: pp 116 - 117] |
| SPS-49 | (5 sec update rate) | [Ref. 3: p 193] |
| APS-138/139/145 | (10 sec update rate) | [Ref. 3: p 221] |

[* Provisional values for prototyping effort. Further, the maximum number of tracks per sensor will be assumed to be 100. Actual values will vary, and are often classified.]

Also note that the rotational speeds of mechanical devices will vary (often nominally) due to mechanical wear, defects, environmental stresses, etc. Rotational values are not constants, but bounded averages. For example, the APS-145 may rotate approximately once every ten seconds (deviations due to mechanical considerations).

***b. Asynchronous Systems***

(1) Man-Machine Interface. Human behavior and performance does not lend itself toward accurate prediction. Notwithstanding, human factors engineers and engineering psychologists have extensively studied human behavior and performance as it pertains to man-machine interaction. An good reference guide on the subject is *Designing the User Interface*, by Ben Shneiderman [Ref. 21].

Within a military context, MIL-STD-1472D "Human Engineering Design Criteria for Military Systems, Equipment, and Facilities" provides useful guidelines and requirements for computer performance in response to human operators. Table XXVIII from MIL-STD-1472D is provided (see Table 1) to indicate recommended system response times. [Ref. 29: p 264]

| Dialogue Type | Required User Training | Tolerable Speed of System Response |
|---|---|---|
| Question and Answer | None | Moderate (0.5 to less than 2 secs) |
| Menu Selection | None | Very Fast (less than 0.2 secs) |
| Form Filling | Moderate | Slow (greater than 2 secs) |
| Function Keys | Moderate | Very Fast (less than 0.2 secs) |
| Command Language | High | Moderate/Slow (0.5 to greater than 2 secs) |
| Natural/Query Language | Moderate | Fast (0.2 to less than 0.5 secs) |
| Graphic Interaction | High | Very Fast (less than 0.2 secs) |

**Table 1. Dialogue Type Versus User Training and System Response**

For calculation purposes, human typing speeds range from 20 - 500 keystrokes per minute. The average English word is approximately 5 characters in length. [Ref. 23: p 335.]

(2) Communications Links. While many characteristics of a given communications system are straightforward and easy to measure (e.g., transmission baud,

protocol characteristics, etc.) the actual message traffic itself is asynchronous, and does not lend itself to parameterization.

C3I system performance will degrade as message traffic increases. However, C3I systems can and must be built to degrade gracefully. The user must be able to control, in large measure, the manner in which his C3I system behaves in response to system overloading. By enabling the user to impose message filters and precedence queueing schemes upon message traffic the "most important" messages will still be processed.

Table 2 provides a provisional set of system response times for ingressing and egressing communications messages.

| Message Precedence | Time Between Message Completion and Transmission | Time Between Message Reception and Display |
|---|---|---|
| FLASH | Very Fast (less than 1 sec) | Very Fast (less than 1 sec) |
| IMMEDIATE | Fast (less than 2 secs) | Fast (less than 2 secs) |
| PRIORITY | Moderate (less than 3 secs) | Moderate (less than 3 secs) |
| ROUTINE | Slow (less than 4 secs) | Slow (less than 4 secs) |

**Table 2. Message Priorities and System Response**

(3) Sensor Systems. Asynchronous sensors provide contact information at irregular time intervals. To interface with such systems, the proposed system must either be prepared to receive interrupt updates, or store new information into a buffer that will be polled regularly. For C3I purposes, polling updates should occur once

every second. Sensor information should be entered into the track database within one second after polling.

| | |
|---|---|
| SPY-1 | ("several times per second") [Ref. 3: pp 116 - 117] |
| SLQ-32 | (less than 1* sec) |
| SQS-53C | (less than 1* sec) |

[* Provisional values for prototyping effort. Further, the maximum number of tracks per sensor will be assumed to be 100. Actual values will vary, and are often classified.]

(4) Weapons Systems. In a time of peace, weapon systems are not used, and hence their status does not change aside from occasional system failures and routine maintenance. It seems a waste of computer resources to verify the status of a weapon system even once every ten seconds. However, in a time of war, weapons will be dispensed freely and usually under extreme duress. Situations may arise when the weapon system status should be updated within fractions of seconds.

| | |
|---|---|
| CIWS | (1* sec update rate) |
| MK 86 GFCS | (1* sec update rate) |
| TWCS | (1* sec update rate) |
| MK 116 UFCS | (1* sec update rate) |

[* Provisional values for prototyping effort.]

Provisionally a nominal weapon status update rate would be once every second. This parameter should be made a user defined value.

### 4. Commonality and Standardization

The Generic C3I Workstation, being *generic*, will provide the most common implementation independent C3I functions, while stressing commonality of user interfaces.

Insofar as the specific communications systems interfaces and platform sensor interfaces may vary from one installation to another, the central Generic C3I Workstation hardware and software will, in large measure, remain identical. Commercially available microprocessor-based workstations are compact and may be easily installed on aircraft, ships, submarines and shore bases. By making use of industry standards, these workstations could support the same operating system, and ensure portable software.

The Generic C3I Workstation shall adhere to the latest version of the *Standardization Guidelines for Developing NCCS Afloat Subsystems*. The Generic C3I Workstation shall use tactical display symbology consistent with fleet standards. "The Requirements Analysis for a Low Cost Combat Direction System," (NPS Master's Thesis by LCDR James A. Seveney and LCDR Guenter P.Steinberg [Ref. 19), is a good unclassified source for standard naval display symbology. The Generic C3I Workstation shall use a man-machine interface consistent with the most recent version of the *Software Requirements Specification for the NCCS-A Workstation Executive, Volume 1: Man-Machine Interface*.

## C. SYSTEM GUIDELINES

While not constituting bonafide goals or constraints, the following guidelines serve as general principles that should be adopted by the development team. In general, these principles represent good programming practices.

### 1. Improved Performance

The performance constraints for the generic C3I workstation include hard-real-time information processing and display. With today's technology, it is possible to implement such a system on a 100 MIPS class machine. Presently, operational systems use dissimilar technology from the hardware envisioned in this study. While *optimal* performance is not necessary during the prototyping effort, a policy of "the faster the better" should be adopted while evaluating algorithm and system-related performance.

### 2. Modular Design

Throughout a rapid prototyping software development project, requirements are changing. The software developers are urged to anticipate change. Thus, software must be constructed in a modular manner, so that changes made to one module or function will minimize the number of changes necessary to other modules or functions. Further, the system software will employ modular design concepts in keeping with an open systems architecture.

### 3. Software Reuse

"'A tentative conclusion is that of all the code written in 1983, probably less than 15% is unique, novel and specific to individual applications. The remaining 85% appears to be common, generic and concerned with putting applications onto computers.'" [Ref. 2: p 5] The turn around times associated with rapid prototyping developments may be correlated with the amount of reusable code available to the development team. "'Software reuse [is] a keystone in many efforts to improve productivity.'" [Ref. 2: p 5]

Developers generating Ada code for the Generic C3I Workstation must make efforts, wherever possible, to create generic reusable software modules. Reusable Ada code will be added to the CAPS reusable software library for future prototyping efforts.

# III. ESSENTIAL MODEL OF A GENERIC C3I WORKSTATION

## A. THE ENVIRONMENTAL MODEL (EXTERNAL INTERFACES)

The three components of a Yourdon Environmental Model [Ref. 26, p 333 ff] include a statement of purpose, a context diagram, and an event list. These three elements are provided in order.

### 1. The Statement of Purpose

The goal of this effort is to develop the prototype of a hard-real-time Ada software system that provides some of the basic features of a Generic C3I Workstation on a commercially available microprocessor based workstation.

The purpose of the Generic C3I Workstation is to provide commonality and connectivity between naval platforms and land bases by providing the ability to process tactical data from many interfaces in real time. This includes the ability of the C3I workstation to receive command and control data from communications links, to receive track information from platform sensors, to provide a tactical display interface to the user, to provide a form-based syntax-directed editor for generating and forwarding communications messages.

### 2. The Context Diagram

The Generic C3I Workstation (see Figure 12) will provide decision support for the user, enabling him to query information resident in the system, as well as change his tactical display by geography, tracks of interest, and scope.

The Generic C3I Workstation will provide a means for resolving track ambiguities as well as the capability for manual insertion and deletion of track information. While the system will contain an embedded expert system for verifying track data integrity, a human operator will be given the opportunity to verify and validate track information contained within the system.

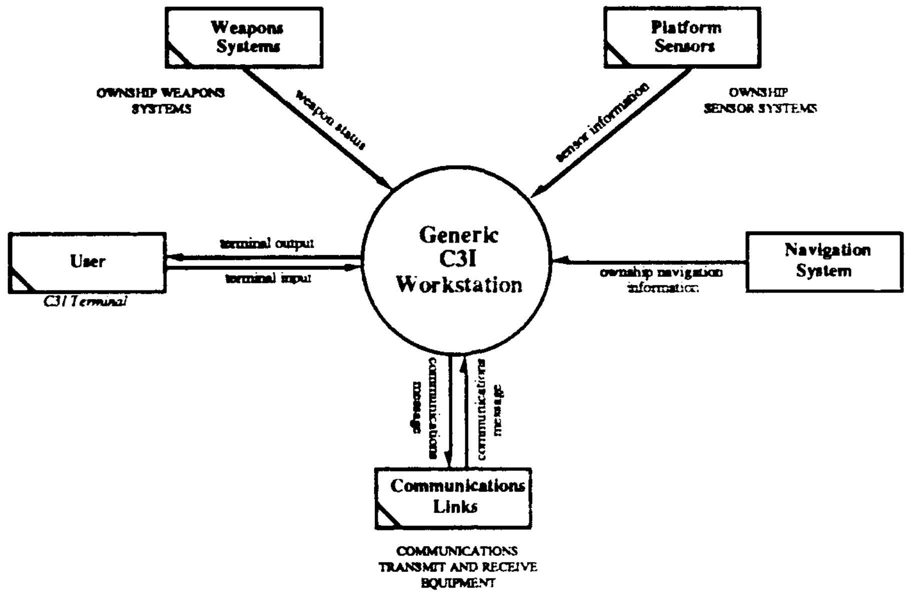



**Figure 12. Generic C3I Workstation Context Diagram**

The Generic C3I Workstation interacts with a number of external systems (or *terminators*). While the system is generic in large measure, the specific external inputs to the system may vary from one instantiation to the next. Provisionally, some of the external interfaces are optional, or may vary in cardinality. As a minimum, the system would require at least one user, and at least one communications link for it to be functional.

***a. Terminators***

(1) User (via C3I Terminal). Whether a user is a Composite Warfare Commander, Officer in Tactical Command, Warfare Area Commander, Tactical Action Officer, Communications Officer, etc., he will primarily be concerned with C3I managerial issues pertaining to information gathering, information integrity, information synthesis and decision making based upon the information presented.

(2) Communications Links. Any digital communication system (encrypted or otherwise) that is capable of transmitting and receiving digital messages, may be connected to the Generic C3I Workstation.

(3) Platform Sensors. Any locally-mounted device that is capable of identifying the azimuth, elevation, range, velocity and/or heading of a contact or track is considered to be a "platform sensor."

(4) Navigation System. A major assumption of the Generic C3I Workstation is that any given platform will be able to provide accurate own-ship positioning, course, velocity and time data. While some older naval platforms currently rely on many different systems to provide this information, the advent of a global positioning system will make this type of accurate information available to nearly every U.S. Navy ship, aircraft, or submarine (with caveats). Shore-based installations are assumed to be immobile, and hence would not need to be provided with periodic navigation updates. In such cases, implementation-specific position information will be provided by a compatible software interface.

(5) Weapons Systems. While not every shore base, aircraft or ship that has a Generic C3I Workstation installed will also be an armed combatant, many U.S. Navy platforms will carry a cadre of weapons. For those Naval platforms that consider own-ship weapon loadout, availability and status to be an important aspect of Command and Control, the Generic C3I Workstation will make this information available to the battle manager.

#### *b. System Input and Output*

This section provides an informal description of Generic C3I Workstation inputs and outputs, focusing upon information content rather than unique representations. A more formal description of data passed to and from the system is included within the data dictionary (cf. Appendix E). Most communications message formats and hardware protocols will be idiosyncratic to the individual systems that are connected to the Generic C3I Workstation. Interface Design Specifications (IDSs) provide the engineering-level detail of data formats and protocols necessary to pass information between operational systems. A top-level description of input to and output from the Generic C3I Workstation follows.

(1) Terminal Input. The set of all valid keyboard keystrokes, audio input, and pointing device selections that may be used to enter data, and interact with the system software.

(2) Terminal Output. The set of all audio or visual responses available to the system, that indicate the termination of a task, the prompting of the user, or the update of currently displayed information. This may include interactive or output windows containing textual or graphical displays.

(3) Communications Message. Abstractly, a communications message could include any discrete packet of information. In general, U.S. Navy communications messages will adhere to strict format requirements. For the purposes of C3I networking, any information passed directly from one computer system to another may be considered to be a "communications message" as well.

(4) Sensor Information. That data provided by specific sensor equipment. As stated earlier, this information will differ from one sensor to another in terms of what information is provided, its accuracy and timeliness. In general, sensor information will include at least the bearing and range of a set of tracks, relative to own-ship.

(5) Own-ship Navigation Information. Navigation information includes: own-ship position (latitude and longitude), own-ship altitude or depth, own-ship course, own-ship velocity, and Greenwich Mean Time (GMT).

(6) Weapon Status. That data provided by specific weapon or weapon fire control equipment. As a minimum, weapon status should include information concerning the weapon's availability and magazine loadout. Provided that a fire control system is capable of indicating which targets are currently assigned to the given weapon system, this information should also be included.

### 3. The Event List

The following is an informal list of events that occur outside of the Generic C3I Workstation prototype and invoke a system response. This list represents a set of generic stimuli that could apply to many specific C3I workstation implementations. Since the Generic C3I Workstation is a small-scale prototype, this event list is a functional subset of an event list associated with full-scale operational C3I station which may include many additional terminators or information sources.

1. Network communications message received (via communications links)
2. Sensor system data update received
3. Weapon system change of status
4. Navigation system updates own-ship navigation information
5. User chooses to view track tuple information (textually)
6. User chooses to manually add new track to database
7. User chooses to manually modify existing track data
8. User chooses to manually delete track from database
9. User chooses to view own-ship weapon status

10. User chooses to view track data (graphically)

11. User chooses to generate a message

12. User enters message text

13. User chooses to send message to addressee

14. User chooses to read a message

15. User chooses to set system parameters: initiate transmission sequence

16. User chooses to set system parameters: set monitor constraints

17. User chooses to set system parameters: archive set-up

18. User chooses to set system parameters: set track filter

19. User chooses to set system parameters: reporting set-up

20. User chooses to set system parameters: network set-up

21. User chooses to set system parameters: emissions control command

## B. THE BEHAVIORAL MODEL (INTERNAL INTERFACES)

A Yourdon Behavioral Model provides an informal means for describing the internal behavior of a proposed software system. Two primary components of a Behavioral Model are the Process Model and the Data Model.

A generic surface-platform installation serves as an example for the subsequent functional decomposition. The rationale behind choosing a surface-platform example was that it possesses examples of all possible terminators, including a navigation system, platform sensors, and weapons systems.

A shore-based installation does not require a navigation system, as its location is fixed. Further, most shore-based command centers do not maintain weapons for their self-defense. Similarly, many airborne-platform installations will not be concerned with weapons systems.

Any given installation may differ considerably in terms of the specific communications links, platform sensors, and weapons systems that are entailed. The underlying principle behind the Generic C3I Workstation system is that regardless of the specific configuration, there is a high degree of functional commonality. The generic C3I workstation attempts to automate the most common implementation independent C3I functions.

Further, the Over-The-Horizon Targeting (OTH-T) Gold Reporting Format [Ref. 30] is a satisfactory choice for examples and prototype development for a number of reasons. First, it is character oriented. Second, messages sent over the OTCIXS link may contain textual message data and/or track-position data. Third, it appears to be the only unclassified character-oriented reporting format in use by the U.S. Navy.

While this thesis uses OTH-T Gold formats, the Generic C3I Workstation supports an open systems architecture and requires a large repository of software modules for processing U.S. Navy communication message formats and interfacing with platform sensors and weapon systems. Reusable components comprise the core of the Generic C3I Workstation system software. If designed and built properly, the underlying structure of the system should never need to change from one instantiation to the next. Only the hardware interfaces and associated message processing software would need to change.

### 1. Generic C3I Workstation Overview

Data flow diagrams have been used to describe the fundamental processes incorporated in the preliminary Process Model for the Generic C3I Workstation. Data flow diagrams offer a flexible means of graphically presenting system functions and their associated data objects. The functional decomposition of the Generic C3I Workstation and a set of associated data flow diagrams are provided in Appendix C. The process specifications for the data flow decomposition are provided in Appendix D. The data dictionary for the Generic C3I Workstation is provided in Appendix E. Appendices C, D and E together provide an abstract model for the Generic C3I Workstation and represent work original to this thesis.

The expanded context diagram of the Generic C3I Workstation (see Figure 13) identifies the primary processes performed by the system. The following six sections correspond with the six processes contained within the expanded context diagram.

#### *a. Communications Interface (Accept, Format & Route)*

The Communications Interface performs those functions directly related to the transmission and reception of communications messages. The implementation of this module may vary greatly from one instantiation of a Generic C3I Workstation to another, due primarily to the fact that U.S. Navy ships, aircraft, submarines and shore-bases are not equipped with the same communications systems. Since the specific interfaces will vary, the implementation of this portion must be highly modularized.

While the specific hardware interfaces vary from one platform to another, the requisite functionality of the Communications Interface will remain consistent. Communications messages that are received will be processed to determine (a) if they contain track information, (b) if they should be forwarded to other communications network participants, and (c) if they contain orders, actions or messages to be brought to the user's attention. Messages will be stored (differentially archived) for future reference.

The Communications Interface will need to monitor, relay and transmit messages on various link networks. Internally, the Communications Interface must perform filtering functions, message routing functions, message precedence sorting, as well as message format translations.

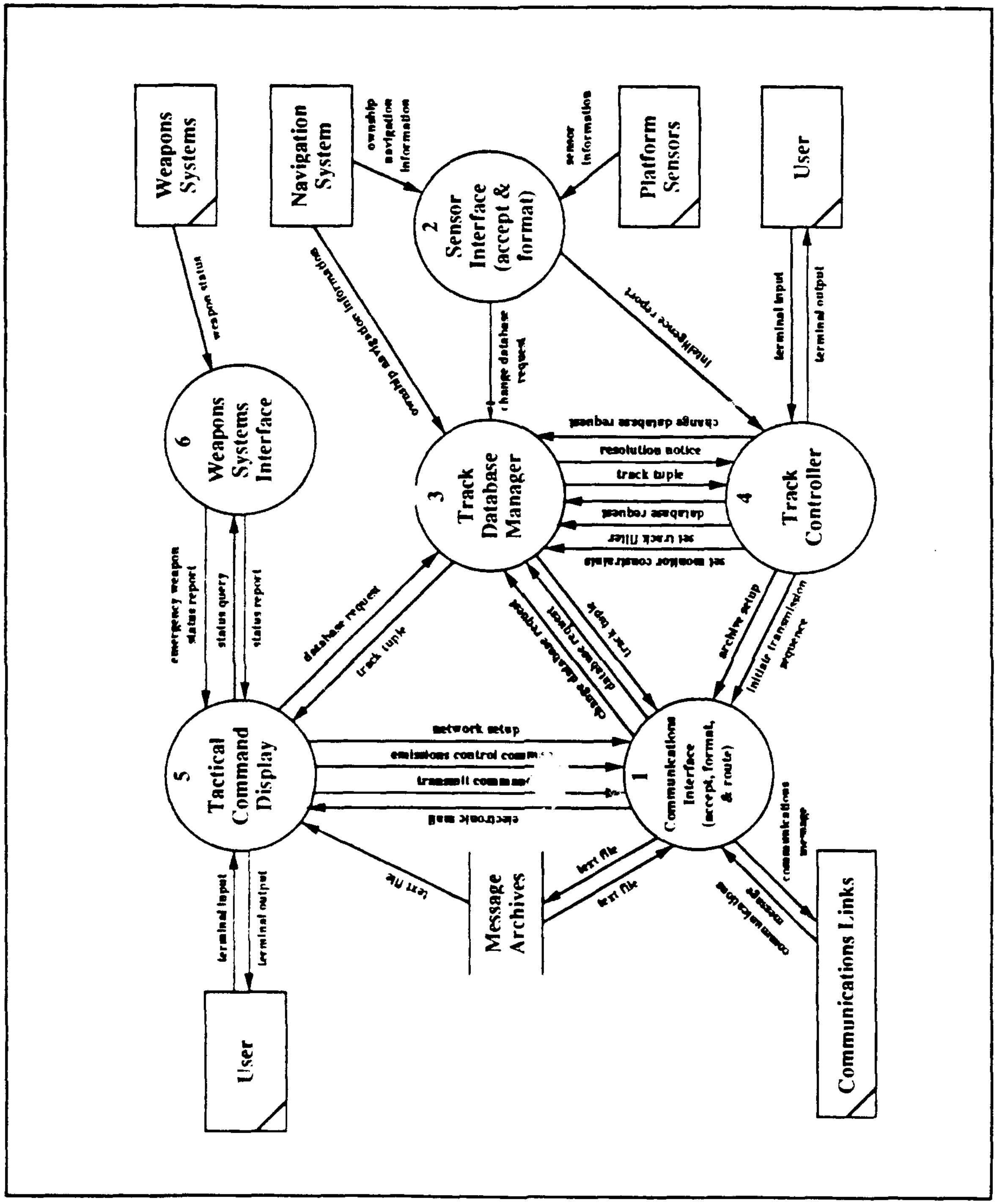



Figure 13. Expanded Context Diagram

In keeping with the provisional shipboard example, the primary communications systems presented are LINK-4A, LINK-11, LINK-16, and OTCIXS. Each of these communications systems maintain separate standards for message formats and hardware interfaces.

### b. *Sensor Interface (Accept & Format)*

There are many different types of sensors used by the U.S. Navy. Some sensors are active, while others are passive. Some sensors provide two dimensional data, others three dimensional. Some sensors are more accurate than others. Some sensors are faster than others. Some sensors are capable of tracking more targets than others. Some sensors provide information periodically, while others provide asynchronous tracking/targeting data.

Because nearly every ship, submarine and aircraft class will be equipped with different sensors, the sensor interface for a Generic C3I Workstation must be highly modular and adaptable.

Sensor information will be received from all "own-ship" platform sensors. The information provided by the given sensor (radar, sonar, optronics, infrared search and track, etc.) will be processed to conform to track related data fields, accounting for data accuracy, word lengths, unique identifiers, etc. New tracks, loss of existing tracks and changes in kinematic constraints would be notified to the central track database manager for possible inclusion into the combined track data store.

In keeping with the provisional shipboard example, the representative sensor systems presented are the SPY-1 radar, SAR-8 IRST, SLQ-32 ESM device, and SQS-53C bow mounted sonar. These systems maintain their own system constraint, data formats, and hardware interfaces.

### c. *Track Database Manager*

The most critical timing constrains pertaining to the Generic C3I Workstation are associated with the real-time storage, update and retrieval of track information. Track data must be constantly refreshed and updated for fleet tactical use. In reflecting upon a system like the Naval Tactical Data System (NTDS), a major criticism is its limited computational abilities, including a limited number of tracks and inadequate processing speed. The primary system trade-offs contrast the number of tracks to be handled against the computational time required to process them. Thus, the more tracks there are in the system, the longer it will take the system to process them. While many technological advances have been introduced since the advent of NTDS, it is not known for certain what information management limitations would exist if state of the art equipment and database systems were used.

Track information received from communications systems and organic platform sensors must be integrated and synthesized by the Tactical Database Manager and combined track data store. The Tactical Database Manager will be responsible for including all new tracks, delete all old tracks, and permitting rapid access to information fields by the user (human operator).

To meet timing constraints, the Track Database Manager will be reasonably indiscriminate about what information will be included into the combined track database. The Track Database Monitor (an expert system) will periodically scrutinize track information and make decisions about what tracks should be matched, merged, correlated or deleted from the database.

#### d. *Track Controller*

As Artificial Intelligence techniques and expert system tools become more reliable, it may become feasible to entirely automate the process of verifying track data integrity. However, as currently envisioned, a human operator is still needed to provide system functions associated with the accuracy and timeliness of track information. The primary tasks of a human operator performing track management functions will be to modify system constraints, to review track information, and add, delete or modify track related information as deemed appropriate.

The Track Controller module must therefore support data queries, information display requests, generation of new tracks, deletion of existing tracks, editing/altering of information in existing track data fields, as well as the initialization of system variables and parameters associated with track data transmissions and battle group level track management.

#### e. *Tactical Command Display*

The primary user of the Generic C3I Workstation will be a commander who is concerned with assimilating track and message data, and how this information will affect a course of actions. The functions provided for the primary user include viewing track data as well as reading and generating communications messages. The Tactical Command Display must therefore support textual message display, textual message editing, information display requests, and graphical track data displays (tactical plots).

The goal of the Tactical Command Display function is to provide the primary user with a robust set of tools for accomplishing his mission. The preferred type of console is one that supports a graphical windowing environment, with iconic function selections. In support of the primary user's task of viewing graphical track data, the

Tactical Command Display must support data information requests, and changes in display requests (areas of interest, tracks of interest, etc.). In support of the primary user's task of displaying and editing communications messages, the Tactical Command Display must provide a robust interactive message generator that will provide templates for administrative message handling, text editing for the actual writing of the body of the message as well as the transmission routing information. Further the message routing function should provide routing transparency whereby predefined addresses and routing protocols will place fewer demands upon the console operator. Wherever possible, the Tactical Command Display must also provide connectivity to platform related local area networks that support message transfer and electronic mail.

#### *f. Weapons Systems Interface*

It is conceivable that some weapons may not provide automated digital weapon status information to a centralized monitor (e.g., consider gun systems aboard battleships which predate digital electronics). However, the Weapons Systems Interface portion of the C3I workstation is designed to monitor current weapons systems status, including operational availability, reload status, weapons loadout, and whatever additional information may be deemed relevant. This information is gathered for the purposes of automating weapon status reporting over communications links. (As an extreme, this module could be expanded to include a full-blown ship status monitor, capable of monitoring not only weapons systems, but also battle damage, compartment flooding, fire and smoke damage, etc.)

While there may be many modern weapon systems that maintain status information, stand-alone weapon systems (e.g., the Phalanx close-in weapon system (CIWS) [Ref. 3: pp 293 - 294]) by definition do not necessarily provide real-time information to any centralized location. However, the following example presupposes that systems have been (or will be) created to support real-time monitoring of all platform mounted weapons systems. In certain cases, this function may be provided manually (i.e., by a terminal and a human operator).

In keeping with the provisional shipboard example, the primary weapons presented will be a PHALANX CIWS, a 5"/54 gun, TOMAHAWK cruise missiles, and Mk 46 torpedoes. Most fire control systems (FCS) were not designed to interface with external systems, and thus maintain idiosyncratic system constraints, data formats, and hardware protocols.

### 2. Process Specifications

To further clarify the lower level processes associated with the Generic C3I Workstation, process specifications are provided in Appendix D. These specifications provide preliminary timing constraints and amplifying comments. Appendix D makes use of a simplified precondition-postcondition format.

# IV. FUNCTIONAL SPECIFICATION

## A. C3I NETWORKING DESCRIPTION

Chapter I provided a brief overview of the U.S. Navy command and control structure. A tiered, multi-layered management scheme serves as the basis for the abstract design of the Generic C3I Workstation.

Commanders may be provided a great deal of autonomy with regards to their own actions. While they may be under operational constraints or *rules of engagement* (ROE), there is still a tremendous amount of flexibility in how a particular commander may choose to execute his responsibilities.

Warfare systems architects and engineers presuppose the wartime possibility that ships, aircraft, and submarines may find themselves operationally or logistically cut off from the rest of the fleet. Because radio transmissions may be jammed or intercepted, battle group actions may have to be undertaken without the benefit of communications. Effective and thorough contingency planning are major issues within modern command and control. However, it is widely believed that effective force level command, control, communications and intelligence will dramatically improve force coordination and potency. The goal of C3I is to maximize warfighting effectiveness while minimizing resource expenditures.

## B. CURRENT C3I NETWORKING APPROACHES

To coordinate the activities of a naval battle force or battle group, an intricate communications structure must be provided. Communications structures will vary from one battle group to another, based upon available platforms and equipment.

For two or more commanders to communicate, they must be able to transmit and receive messages (processing is implied). Communications will be defeated if the transmitter is not operational, the receiver is not operational, or the spectral transmission medium is in conflict (jamming or multiple simultaneous transmissions -- i.e. "collisions").

The obvious shared resources among communications network participants are: time, and transmission media. What passes between network participants are "messages" or units of information. The format or content of messages vary from one network to another depending upon the purpose of the particular network, or the goals of particular network users. As more and more messages are exchanged between network participants, resource conflicts arise. Mission critical messages must get delivered, and yet are occasionally lost or delayed due to causes ranging from human error to equipment failure to poor planning to information overload.

## C. PROPOSED C3I NETWORKING FUNCTIONALITY

The Generic C3I Workstation represents a means for providing communications between network participants. By networking Generic C3I Workstations in support of a Composite Warfare Commander (CWC) Command and Control Architecture, a new set of functionalities becomes apparent.

### 1. Warfare Mission Area Breakdown

"A composite warfare commander (CWC) doctrine is used to enhance the management of these assets in a concerted sea-control effort that coordinates the three dimensional air, surface, and subsurface defense of a battle group." [Ref. 24: p 55] Within a multi-layered task management structure, the individual in overall tactical command would delegate authority to subordinate commanders and coordinators for the purposes of conducting and administering control of forces pursuant to their particular missions. [Ref. 24: p 55]

By providing each of the following individuals with a Generic C3I Workstation the C3I functions that they perform will be facilitated. The specific information maintained by each installation would vary according to need and area of interest. Each user may set local precedences and filters to tailor their particular Generic C3I Workstation to meet individual performance requirements.

#### a. *Composite Warfare Commander*

The Composite Warfare Commander (CWC) or Officer in Tactical Command (OTC), has the greatest authority within a battle group. This individual is responsible for successfully employing individual antisubmarine warfare (ASW), antiair warfare (AAW), antisurface warfare (ASuW) and strike warfare (STW) forces in concerted sea-control efforts. [Ref. 24: pp 54 - 55] The CWC needs to access information from all warfare areas and maintain a complete picture of his battle group asset locations and dispositions to assess and address force pervasive issues. The CWC issues operational force orders down the chain of command, and responds to higher level instructions. [Ref. 31: p 30]

#### b. *Antiair Warfare Commander*

Antiair Warfare (AAW) refers to that portion of sea control associated with the protection of the battle group from the threat of enemy aircraft and missiles. Since direct defense of carrier or battleship battle forces is provided by the battle forces themselves, [Ref. 24: pp 54 - 55] the Antiair Warfare Commander (AAWC) is in charge of coordinating battle force resources to minimize damage to friendly units, and maximize the damage to hostile units.

An AAWC needs to maintain an accurate tactical picture of the air threat, along with friendly force location and disposition with regards to accomplishing the AAW mission. To perform his task well, AAWC must be capable of achieving early warning of hostile actions and accurate information pertaining to the management of all systems under his command. [Ref. 4: p 75] The AAWC must respond to orders from higher authority, coordinate with other warfare mission area commanders over the use or deployment of shared battle group assets (fixed wing aircraft, and AAW capable surface combatants), delegate authority, and issue tasking orders to subordinate commanders.

### *c. Antisubmarine Warfare Commander*

Antisubmarine Warfare (ASW) uses sea control and sea-denial forces to protect the battle group from submarines and underwater threats imposed by hostile forces. [Ref. 24: p 55] "The [Antisubmarine Warfare Commander (ASWC)] is traditionally thought to be interested in screen composition, screen size, limited lines of approach for a given subsurface threat, and other defense-oriented ASW functions." [Ref. 27: p 81]

The ASWC must maintain an accurate tactical picture of the subsurface threat, along with friendly force location and disposition with regards to accomplishing the ASW mission. The ASWC needs to be able to respond to orders from higher authority, to coordinate with other warfare mission area commanders over the use or deployment of shared battle group assets (ASW capable ships, fixed wing aircraft, helicopters, and submarines), to delegate authority, and to task subordinate commanders.

### *d. Antisurface Warfare Commander*

Antisurface Warfare (ASuW) refers to offensive sea-denial missions, focusing upon the destruction of enemy ships. The Antisurface Warfare Commander (ASuWC) plans and coordinates actions that must be taken by friendly forces to destroy targets effectively and efficiently. The ASuWC may command the use of cruise missiles or fixed wing aircraft to accomplish his missions. [Ref. 27: p 81]

The ASuWC must maintain an accurate tactical picture of all shipping traffic (neutral, friendly, or hostile, commercial or military) within the range of weapons at his disposal to effectively conduct mission planning. This implies the need to maintain an accurate, timely, and complete over-the-horizon track database. [Ref. 28: pp 85 - 86] The ASuWC needs to be able to respond to orders from higher authority, to coordinate with other warfare mission area commanders over the use weapons or the deployment of battle group assets (ships, fixed wing aircraft, and helicopters), to delegate authority, and to task subordinate commanders.

#### *e. Strike Warfare Commander*

Strike Warfare (STW) refers to offensive actions taken against enemy land targets. The Strike Warfare Commander (STWC) attempts to maximize damage to designated targets at minimum cost. The STWC has a dazzling array of modern ships, aircraft and assault craft at his disposal for conducting strike operations. [Ref. 4: p 74]

In keeping with his power projection role, the STWC must maintain an accurate tactical picture of designated targets, enemy defensive installations, as well as friendly force location and disposition. The STWC must be capable of responding to orders from higher authority, coordinating with other warfare mission area commanders over the offensive use or deployment of battle group assets (fixed wing aircraft, helicopters, cruise missile launching platforms, and naval surface gun fire support ships), as well as delegating authority and tasks to subordinate commanders.

#### *f. Force Coordinators*

Because a battle group is a very complex organization, the CWC may designate various commanders to ensure proper coordination and interoperability within a given battle group. Coordinators would respond to orders from higher authority and interact with warfare mission area commanders over the cooperative use of battle group assets.

(1) Force Over-the-horizon Track Coordinator. "The [Force Over-the-horizon Track Coordinator (FOTC)] performs three vital functions for the battle group. First, he maintains a surface and subsurface database that includes both potential threats and friendly surface traffic, to ensure accurate targeting. Second, the FOTC is a vital link in monitoring the flow of non-organic intelligence information to the battle group for generation of new tracks and for updating and correlating new data. Finally he provides targeting data for all battle group war-at-sea strikes." [Ref. 27: p 79]

(2) Electronic Warfare Coordinator. The Electronic Warfare Coordinator (EWC) monitors and controls electromagnetic emissions produced by the battle group. The EWC attempts to minimize the adversaries electronic warfare capabilities through the coordinated use of the electromagnetic spectrum. [Ref. 27: p 81] In a transiting mission, the EWC may conceive of elaborate cover and deception (counter surveillance) ploys based upon selected use of electromagnetic emissions. In a hostile environment, the EWC may ensure that battle force electronic counter measures (ECM), and electronic counter counter measures (ECCM) techniques are utilized effectively in defense of the battle group.

### 2. Network Track Database

Each Generic C3I Workstation is fully capable of maintaining its own track database. Autonomous operation is an important feature in the advent of network "casualties." When a network is fully functional, track information management may be relegated to a centralized force track coordinator (cf. the Force Over-the-horizon Track Coordinator). Under such conditions, every unit would be required to periodically forward copies of their track database to that designated individual. The track coordinator would be tasked with consolidating (matching, merging and correlating) the track data from all network participants and then periodically distribute official/sanctioned/approved sets of track data. (See Figure 14.)

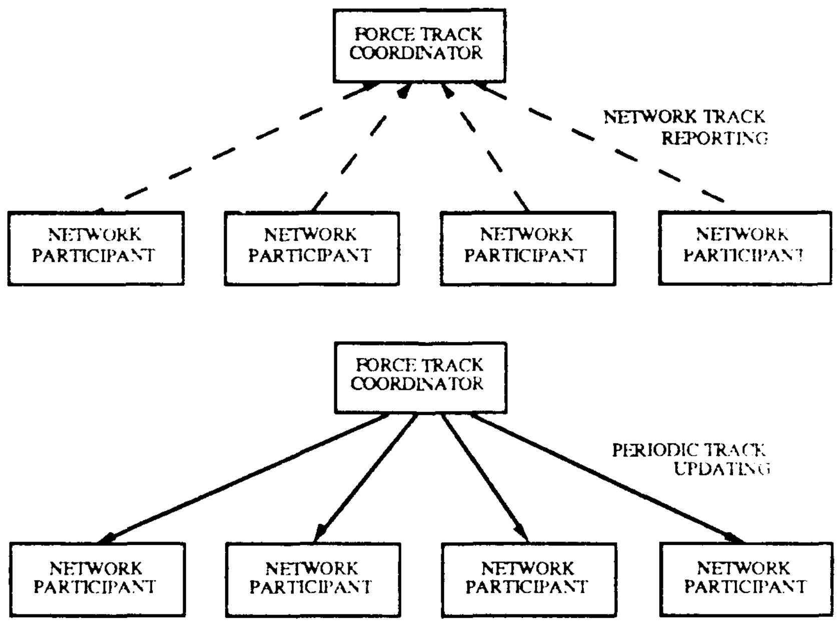



**Figure 14. Centralized Network Database Manager**

While using a centralized database administrator, it should be noted that there will be time delays associated with data collection, data transmission, data processing, data transmission and data assimilation. As currently envisioned, the Generic C3I Workstation would not replace existing communications hardware (transmission and reception equipment). Hence, any deficiencies inherent within a set of communication devices would still remain a limiting factor to the overall performance of a network-distributed database.

Network participants would combine the official track data with whatever local track information is provided from local platform sensors. Since track tuples would include an "origin" attribute, information provided by the FOTC would be noted as such. The user may then display any subset of track information desired: FOTC tracks, FOTC plus local tracks, etc.

### 3. Deadlocks

The term "deadlock" refers to a state within a finite state machine from which there is no exit. Either the final state is not an accepting state, or system control rests in a cycle with no accepting states. In the vernacular of computer science, the system either "hangs" or appears to be in "an infinite loop."

Deadlocks must be avoided. Within a network of automated communications equipment, software must be designed to recognize and avoid possible deadlock situations.

#### *a. System Deadlocks*

As mentioned earlier, the resources shared between communications devices are transmission media and time. Whenever two or more network participants attempt to simultaneously transmit messages over the same media (and frequency), a data collision occurs. If the transmission equipment is capable of performing collision detection, the given communications protocol may require immediate (or time delayed) re-transmission of the queued message. Several communications protocols, including ALOHA (pure and slotted), CSMA (persistent and non-persistent), and CSMA/CD suffer from a statistical possibility that a given message may never be successfully transmitted. [Ref. 22: pp 121 - 130]

#### *b. Functional Deadlocks*

Aside from actual (physical) system deadlocks, there are a plethora of potential functional deadlocks arising through the use of automated communications devices. If mission critical messages are not received or responded to in a timely manner, what would a C3I workstation be expected to do? Today, most communications messages are manually generated, manually read, and (when necessary) manually responded to. Human operators are relied upon to ensure that network information flows properly.

(1) One Way Communications. Many naval communications messages are "one way" in nature. Information is forwarded from a sender to an intended recipient. No acknowledgement is given or required. Figure 15 indicates the simplicity of such a communications scheme.


**Figure 15. One Way Communications**

Situation reports, periodic updates, and routine memoranda, while potentially important in and of themselves, are considered to be expendable. If a particular ship fails to report on time, life will continue. These messages may be considered "for the recipient's information, no reply necessary." Operationally, some platforms may choose to keep electromagnetic silence. Further, ships may be equipped with "receive only" equipment. [Ref. 3: p 12] Information may be gathered, but transmissions are not possible. Hence, whether communications are one way by design or decision, one way messages cannot lead to deadlocks.

(2) Communications Dialogue. In the previous section, if a ship failed to report on time, a specific query may be sent to that vessel, demanding an answer to the question "why did you fail to report?" Such an inquiry represents a classic dialogue. A reply is expected.

In combat situations, force orders will need to be acknowledged by the appropriate recipients. Figure 16 depicts a simple feedback mechanism. Messages demanding a response or acknowledgement may cause the user to delay certain actions until the appropriate response is rendered and forwarded. These delays may potentially lead to deadlocks.

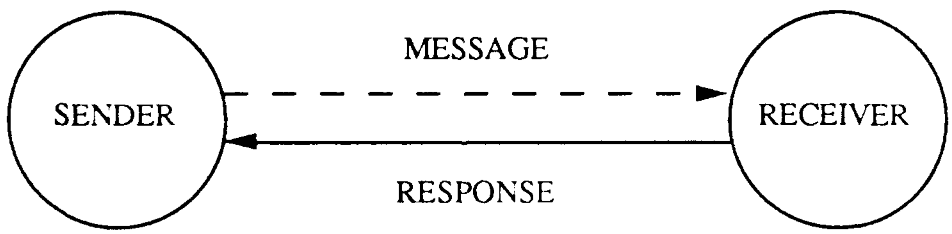



**Figure 16. Communications Dialogue**

A major drawback of manually generated communications is that the sender of a message may have to wait an indefinite period of time before a response is received. Higher precedence responses, distractions, and human error all contribute to turning dialogues into unintentional monologues.

### c. *Automated Message Accounting*

As the level of technology becomes more sophisticated, it becomes possible to flag communications messages as "receive only" or "response required." The system software of the Generic C3I Workstation could provide the user with an additional visual or audio cue for messages that require a response. An additional window could appear that provides the user with the ability to remit "will comply" (WILCO) or "cannot comply" (CANTCO) messages with appropriate amplifying text. If the user fails to respond to the message within 30 seconds, the system may produce an audio tone as a reminder. If the user continues to fail to respond, the system could persist in reminding the user that a reply is necessary (perhaps escalating in volume, number, or duration of audio tones).

Since the Generic C3I Workstation provides multiple functions (i.e., tactical display, text editing, weapon status display, and database functions), it would be inappropriate to cause the entire system "hold" until appropriate human response is provided. It would be inadvisable to remove system control from the user at any time.

## 4. Failure Modes

Occasional external factors may force a change in the behavior of a Generic C3I Workstation. Broadly, any hardware or equipment failure (or damage) directly affecting any system connected to a Generic C3I Workstation could be called a casualty state. The Generic C3I Workstation must be capable of operating in a variety of casualty states. Indeed the utility of such a workstation would decline tremendously if it had no functioning communications links, yet even if a Generic C3I Workstation is completely isolated, it may still provide some degree of local functionality (e.g., weapon status display, tactical picture based on platform sensors, etc.).

### a. *Workstation Degradation*

The Generic C3I Workstation reacts to its environment. If information is provided to the system, then that information is assimilated into the appropriate data stores. If no information is provided to the system, the information is not assimilated.

If a particular set of communications equipment were to "go down," then the system should simply acknowledge the loss and continue functioning with whatever additional communications devices are operational.

Similarly, if a particular weapon system were damaged or experienced a failure, the Generic C3I Workstation should acknowledge the loss and continue to provide whatever weapon status information is available.

Again, if a particular platform sensor system were to be destroyed or suffer mechanical failure, the Generic C3I Workstation would acknowledge the loss and continue to collect information from the other (functioning) sensor systems. The system should be capable of selectively filtering out spurrious information from faulty (or damaged) sensors.

While the loss of a navigation system poses unique difficulties, two possible solutions exist. First, the user may manually insert and update navigation information. This would detrimentally affect track accuracy. Second, the user could make a request from another vessel in close proximity to periodically send track information with regards to own-ship.

### b. *Network Casualties*

If a Generic C3I Workstation itself were to fail, as currently designed, the network would need to be notified of its loss, or deduce its absence. If all network participants were required to send periodic "system up and operational" messages, then the immediate absence of such a message would imply its loss.

The arrival of new network participants and the relocation of others outside of an immediate sphere of operations will be commonplace. On-line programming techniques could assist in the constant evaluation and re-evaluation of network participants. Additional "new network participant" and "quitting network" messages could facilitate the networking process.

The unexpected loss of a network participant is not necessarily crucial, except for the case where the casualty was someone who performed a necessary function. In assessing the CWC C2 Architecture, the loss of the CWC would necessitate the "alternate CWC" to take over the duties of the CWC. Similar re-assignment of network tasking would need to occur in the advent of the loss of a warfare mission area commander, or force coordinator. It is presumed that an alternate commanders or coordinator would have their Generic C3I Workstations set up to receive information addressed to themselves as well as the principal commander or coordinator. Such a back up scheme would permit the alternate commander or coordinator to take over the functions and responsibilities of the principal commander or coordinator in a minimal period of time.

## D. MODIFICATIONS TO CURRENT NAVAL MESSAGES

Any automated C3I network will require unique protocols and identifiers. In summarizing the ideas and concepts presented above, current naval communications messages do not lend themselves well to an automated C3I workstation network. Additional message formats would need to be added to the current operational specifications for naval communications systems.

In the support of a CWC C2 Architecture, new messages would include:

• NETWORK MESSAGE -- a message that identifies either the presence of a new network participant or the departure of a current network participant.

• POLLING MESSAGE -- a message that requests a response from a particular network participant for the purposes of verifying that they are "up and operational."

• DATA UPDATE MESSAGE -- a message in support of battle group database management, where updated information could be forwarded to a designated track coordinator for processing, as well as providing a means for the battle group track manager to rapidly send clean updated track data. (This may go beyond current communications systems).

• DIRECT DATA TRANSFER -- a message from one Generic C3I Workstation to another, which the system may understand or interpret directly without having to process it as a conventional communications message.

In addition, those naval messages that currently do not have an indication or a flag that identifies them as requiring a response would need to include such a mechanism so that the Generic C3I Workstation may avoid deadlocks more effectively.

By providing such messages, an automated (or largely automated) C3I system could be developed. Such a system would be self-adapting based upon the presence or absence of designated commanders and coordinators, resilient to system failures and battle damage, as well as capable of providing automated support currently unavailable to the fleet today.

Caution must be urged in the development of implementation-specific designs for messages. Should the U.S. Navy adopt a C3I architecture alien to the CWC Command and Control Architecture, the system should be able to adapt, with the minimum of effort.

Hence, extensible message formats are recommended, whereby new message types may be added without detrimentally affecting the processing of existing message types.

## E. INITIAL FUNCTIONAL SPECIFICATION

The preceding English narrative describes in an informal manner the functionality of a Generic C3I Workstation. Appendix B provides an initial cut at a formal description of the Generic C3I Workstation. This functional specification makes use of SPEC specification language developed by Dr. V. Berzins of the Naval Postgraduate School.

A formal functional specification provides a more precise (mathematical) description of a proposed system. A formal functional specification not only clarifies the system design, but it also serves as an input for automated software engineering tools (e.g., software debugging tools, verification and validation tools, automatic code generation, etc.). Appendix B provides an initial a top-level description of a Generic C3I Workstation that will provide a framework for future clarifications and modifications.

# V. IMPLEMENTATION MODEL

## A. PROTOTYPING EFFORT

C3I systems demand efficient computation and reliable real-time behavior. Sophisticated C3I systems are difficult to construct and are costly to develop. Consistent with the Navy's Next Generation Computer Resources (NGCR) program, experimental rapid prototyping of a Generic C3I Workstation on commercial, microprocessor-based workstations can demonstrate a low-cost approach to providing the U.S. Navy with affordable and effective C3I systems. The functional description and initial requirements of the Generic C3I Workstation provided in the preceding chapters will serve as the basis of a prototype system that makes use of the PSDL prototyping language and its computer aided prototyping system (CAPS).

### 1. Prototype System Description Language

> The Prototype System Description Language (PSDL) was designed to serve as an executable prototyping language working at a specification or a design level. PSDL is a language for describing prototypes of large software systems with real-time constraints on different levels of abstraction. Such systems are modeled in PSDL as networks of operators communicating via data streams, using augmented data flow diagrams. The operators in an augmented data flow diagram are supplemented with timing constraints and non-procedural control constraints. The data streams can carry data values of an abstract data type as well as tokens representing exception conditions. Each type or operator is either composite or atomic. Composite operators are implemented by decomposing them into networks of more primitive operators using PSDL. Atomic operators are realized by retrieving reusable components from the software base which meet the specifications of operators and are implemented in some programming language. The language is easy to use because it provides a familiar graphical notation for the underlying computational model. A specification which augments a data flow graph provides the information to effectively retrieve reusable software components and adapt them to the specific application context. [Ref. 9: p 2]

This thesis has used existing requirements analysis tools and techniques to develop the functional specification of a Generic C3I Workstation, including data flow diagrams (cf. Appendix C) and a data dictionary (cf. Appendix E). However, PSDL requires additional timing and control constraints.

#### *a. Timing Constraints*

PSDL enables the software developer to easily specify the performance requirements associated with a particular function or software module. Hard-real-time systems require explicit timing constraints. PSDL supports three types of timing constraints: maximum execution time, maximum response time, and minimum calling period. [Ref. 11: p 23]

In formalizing the process specification for a Generic C3I Workstation, timing constraints also needed to be addressed. In Chapter II many high-level system timing constraints were identified. However, the these system-level timing constraints do not readily map onto the subsystem-level data flow diagrams provided in Appendix C.

Consider the requirement, "From the time a track data message is received by the system and its contents are entered into the track database shall be less than two seconds." The two second timing constraint does not directly refer to any single bubble within the data flow diagram. Instead, data must flow between a sequence of processes until the final set of track tuples are included into the track database. Thus, the two second constraint applies to the entire process sequence.

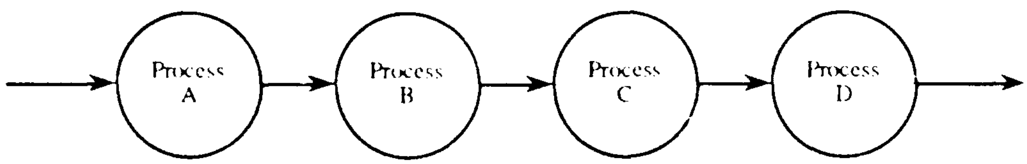



**Figure 17. Process Sequence**

Figure 17 presents a four process sequence. The goal of the prototype system developer will be to insure that the sum of the maximum execution times (met) associated these processes will be less than or equal to the imposed sequence timing constraint.

$$\text{n-process sequence timing constraint} \geq \sum_{i=1}^{n} \text{maximum execution time for process } (i)$$

For the sake of example, suppose that the timing constraint of the process sequence for Figure 17 was two seconds. We may readily deduce that the following must be true.

$$0 \leq \text{met(A)} + \text{met(B)} + \text{met(C)} + \text{met(D)} \leq 2 \text{ seconds}$$

Yet, the software developer must still decide how to allocate the individual maximum execution times.

Without better direction, the developer may arbitrarily assign equal values to all processes within the given sequence. In this example, the developer may assign maximum response times of 500 ms to each of the four processes (i.e., $0 \leq 4$ (500 ms) $\leq 2$ secs). However, this simple allocation of timing constraints overlooks many considerations. For instance, if Process A is an inherently more more complex (and hence slower) process than Process C, then it may make sense to assign a timing constraint of 750 ms to Process A and only 250 ms to Process C (i.e., $0 \leq 750$ ms + 500 ms + 250 ms + 500 ms $\leq 2$ sec).

Currently, timing constraints associated with subsystem level processes must be determined by the expert opinions of experienced programmers and system developers. While the Delphic approach is used to initially assign somewhat arbitrary timing constraints, rapid prototyping will provide empirical results that will lead to more appropriate or realistic timing constraints. Hence, the timing constraints that appear in

Appendix D are baseline values, and are expected to serve as recommendations rather than requirements.

### *b. Control Constraints*

"The control aspect of a PSDL operator is specified implicitly, via control constraints, rather than giving an explicit control algorithm. There are several aspects to be specified: whether the operator is periodic or sporadic, the triggering condition, and output guards." [Ref. 11: p 17]

Control constraints for the Generic C3I Workstation are provided within the process specifications in Appendix D. All processes are sporadic unless clearly stated otherwise. Triggering conditions for operators are stated in the preconditions. Output guards and error constraints are left for the PSDL implementors to develop based upon the data dictionary in Appendix E.

## 2. Computer-Aided Prototyping System

Computer-aided support of PSDL will be provided by an integrated prototyping environment which assists the designer in iteratively constructing a PSDL design and automatically links it to reusable components in the software base. When complete, the computer-aided prototyping system (CAPS) will consist of three primary subsystems: a user interface, an execution support system, and a prototyping software base.

The user interface will contribute to effective and efficient construction or modification of prototypes by providing a graphical editor, a syntax directed editor, a browser, an expert system for communicating with end users, and a debugger. These editors will provide convenient entry and management of PSDL descriptions, the browser will allow the designer to interact with the software database while retrieving and examining prototype components. The expert system will provide a paraphrase capability generating English text from PSDL descriptions. The debugger allows the designer to interact with the execution support systems.

The execution support system will consist of a translator that generates code to link reusable software components together, a static scheduler that allocates time slots for prototype components prior to their execution, and a dynamic scheduler that allocates free time slots to non-time critical components as execution proceeds.

Program construction is sped up by taking advantage of reusable software components drawn from a software base. The aspects of program construction that will benefit most from mechanical assistance are software base retrievals from the software base, generation of code for interconnecting available modules, and static task scheduling. The prototyping database consists of a design database, reusable software base, software design management system and a rewrite subsystem. The prototyping database keeps track of designs and stores reusable prototype components together with their specifications. Its design management system provides version control and maintains design histories, a rewrite subsystem translates PSDL specifications into a normalized format to ease retrieval. [Ref. 9: p 2] (Also see [Ref. 12].)

The Generic C3I Workstation prototype will represent the first large scale prototyping effort to make use of portions of the CAPS system. CAPS has been under development for several years, and is reaching the point where it may be used to automate portions of the prototyping effort.

## B. IMPLEMENTATION CONFIGURATIONS

### 1. Prototype Implementation Model

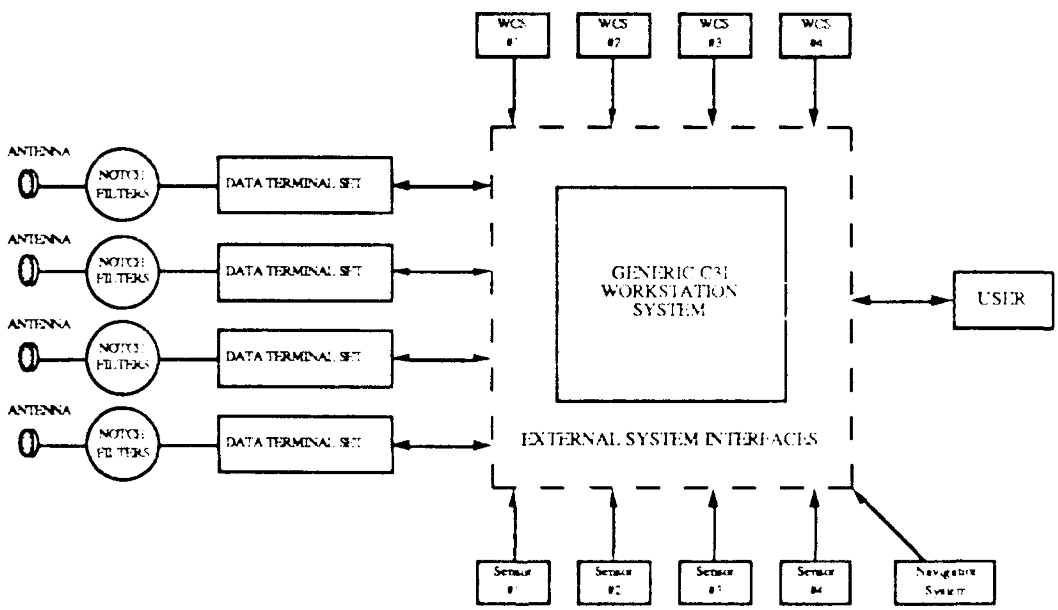



Figure 18. Generic C3I Workstation with a Single-user and External Communications Links

Initial prototyping efforts will focus on a single user system with multiple weapons, sensors, and external communications (see Figure 18).

#### a. *Prototyping Hardware*

The Operational Requirement for Next Generation Computer Resources (NGCR) states that a "family of Navy standard, militarized computers is the most cost effective, efficient means to meet ... [the Navy's] information processing and combat needs." [Ref. 33: p 6]

The Naval Research Advisory Committee on NGCR has "found that both ruggedized and fully militarized versions of commercial computer architectures are available on today's market." [Ref. 33: pp 12 - 13] Among them, Genisco manufactures a ruggedized version of the Sun Microsystems workstation.

The Generic C3I Workstation will be developed using a non-ruggedized Sun Microsystems workstation operated by the Naval Postgraduate School's Computer Science Department. Once the prototype software has been developed, it may then be transferred to ruggedized workstations.

#### b. *Prototyping Software*

The operating system used by the Sun Microsystems workstation (SunOS 4.0) is derived from UC Berkeley Version 4.3BSD and AT&T System V Release 3.2 [Ref. 34: p 3]. Unix™ is an industry-standard multi-user computer operating system. Unix™ offers portability, and supports the use of a windowing environment.

Source code developed by the prototype effort shall be written in the Ada programming language, in accordance with Department of Defense directives. TAE+, a windowing software package written in Ada and developed at NASA's Goddard Space

Flight Center, will be used to generate the user interface for the Generic C3I Workstation prototype.

### 2. Configuration Extensions

A specific instantiation of a Generic C3I Workstation will be interfaced with a limited set of external systems. If a Generic C3I Workstation is to be installed aboard a patrol aircraft, it may only have a single user terminal, interface with only one or two communications links, interface with a single radar system, and not interface with any weapon systems at all. A shipboard instantiation of a Generic C3I Workstation may well be interfaced with four or more external communications systems, four or more platform sensors, and perhaps six or more weapon systems. A shore based instantiation of a Generic C3I Workstation may be devoid of weapons, sensors, and navigation systems, and only process information provided by external communications systems. Regardless of these different configurations, a substantial amount of the software for the Generic C3I Workstation is reusable. Such are the potential benefits of an open system architecture.

Additional thought should be given to developing a multi-user system (See Figure 19). Since many of the functions of a Generic C3I Workstation are independent from one another, a multi-user system may not slow the system down sufficiently to violate real-time constraints, especially in configurations with separate (or multiple) CPUs for each user.

A multi-user system better supports warfare mission area commanders. For example, consider the Anti-Submarine Warfare Commander (ASWC).

> The destroyer squadron commander (ComDesRon) is traditionally assigned as the ASWC. The ComDesRon's small staff consists of seven to nine surface warfare specialists and a single aviator. They are responsible for planning a complex battle program over a large area, which involves many types of ASW forces. This staff may be embarked in the carrier or a battle force destroyer. [Ref. 24: p 55]

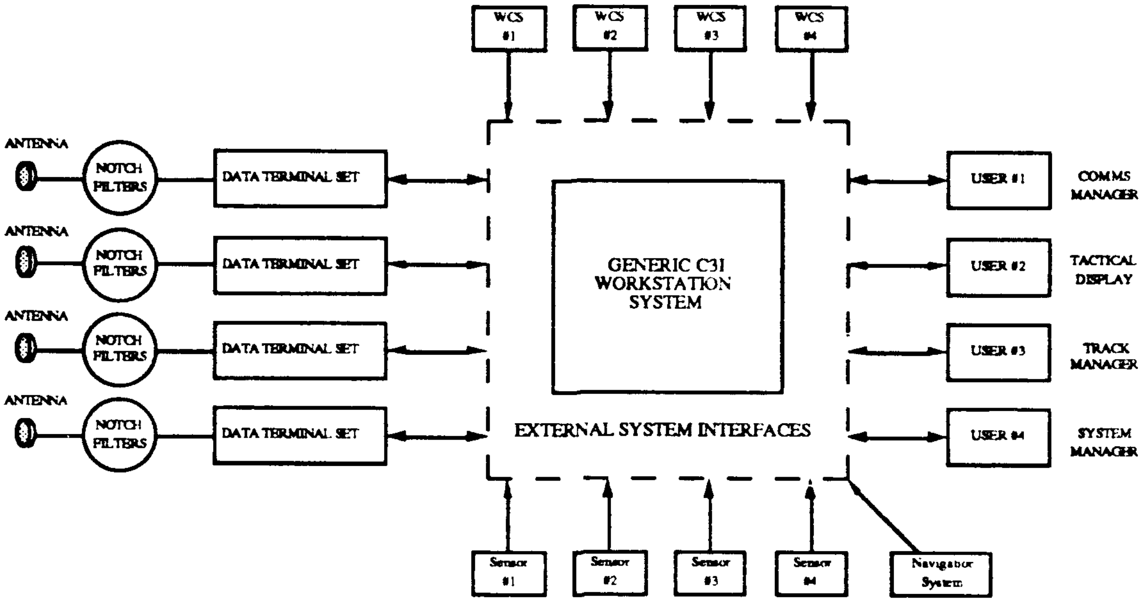



**Figure 19. Generic C3I Workstation with Multiple Users and External Communications Links**

In order to provide C3I support functions to a staff of perhaps a dozen users, the Generic C3I Workstation will need to provide multi-user functionality. A multi-user Generic C3I Workstation will simultaneously enable different users to view tactical situation displays, to generate communications messages, or to read and review orders. An operational Generic C3I Workstation should be capable of supporting a multi-user environment.

Additionally, Generic C3I Workstations may be instantiated aboard the same platform (see Figure 20). Co-located Generic C3I Workstations may be connected via direct data links. Since the NGCR effort also supports the Navy's use of high-speed (100 megabit per second) fiber-optic networks, various Generic C3I Workstation system resources may be tied into the same Survivable Adaptable Fiber Optic Embedded Network (SAFENET). [Ref. 33: p 52]

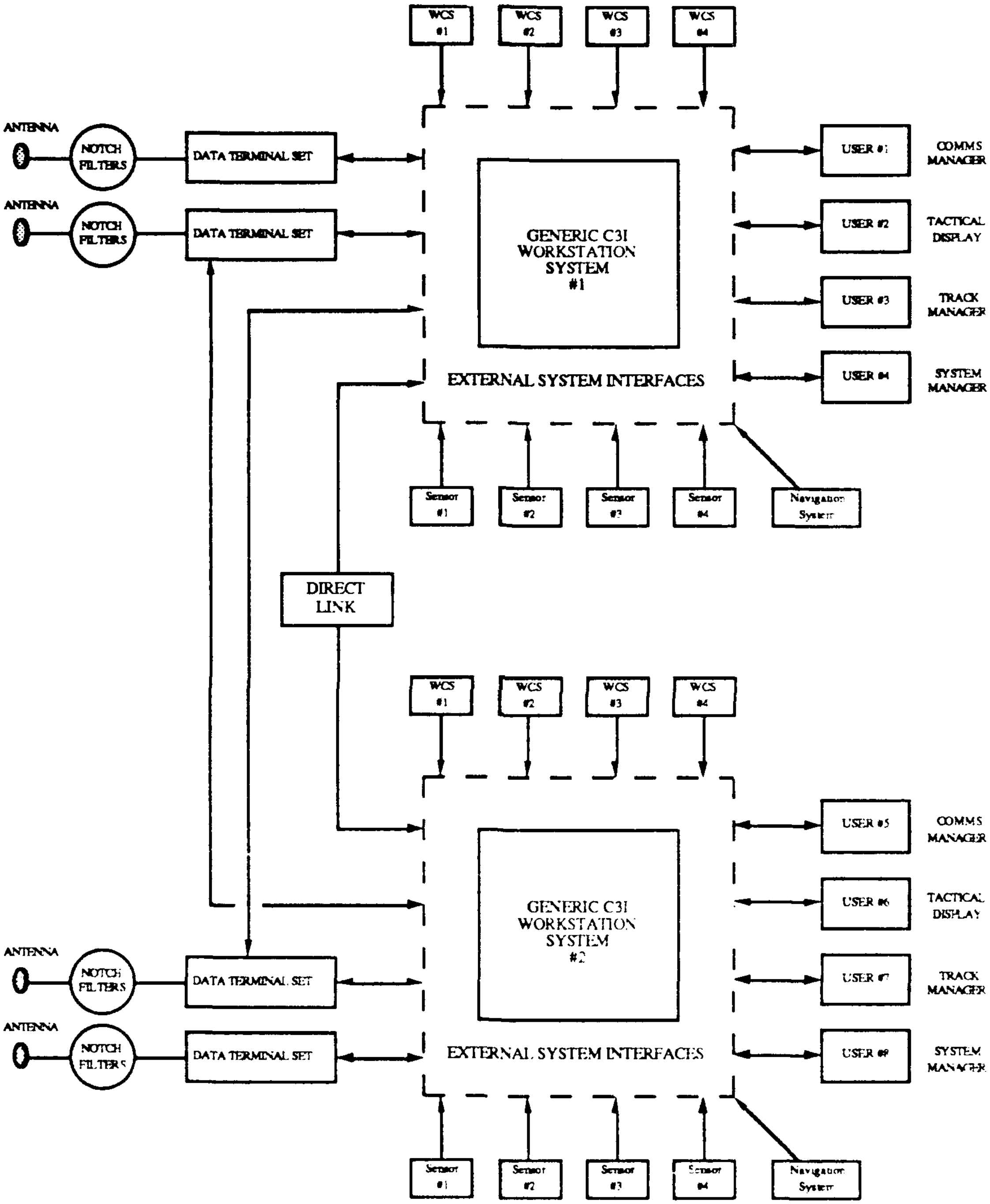



**Figure 20. Locally Networked Generic C3I Workstations With Shared External Communications Links**

With multiple warfare mission area commanders and force coordinators located aboard the same platform and local area networks approaching very high speeds, new sophisticated resource sharing techniques could be developed.

Indeed, the future appears bright for the networking of distributed systems. Not only could Generic C3I Workstations be directly networked, but Generic C3I Workstation Systems themselves could contain a number of microprocessors. Potentially, every external system interface could be controlled by a dedicated central processing unit (CPU). Computationally intensive independent processes could be migrated to separate CPUs (e.g., graphics displays, track database, message translation, communications network monitoring, etc.) in order to enhance the system's overall performance.

In recent years, a number of parallel computing systems or designs have become industry standards (e.g., Ethernet™ as IEEE 802.3). Some of the parallel computer systems are also commercially available (e.g., Alliant Sequent, BBN Butterfly, etc.). Within a few years, parallel systems such as these may become new NGCR standards. Hardware and software parallelism must be anticipated.

# VI. CONCLUSIONS AND RECOMMENDATIONS

## A. SUMMARY AND CONCLUSIONS

The integration of formal requirements with rapid prototyping appears to offer a means to decrease both development time and costs associated with software development associated with C3I systems. While a great deal of requirements analysis is still needed to define a software system, less time is spent in formalizing preliminary requirements for software modules that are anticipated to change during the iterative prototyping cycles.

The major emphasis of the Generic C3I Workstation is to support C3I information management functions such as multiple sensor information correlation, message generation and information display. The Generic C3I Workstation also serves as a gateway between different communications links, for improved connectivity between naval C3I stations. By automating many of the tasks performed by human operators today, more accurate and timely communications may be realized.

By imposing hard-real-time constraints upon the Generic C3I Workstation's information processing, the user is provided with real-time (or near real-time) tactical information. Accurate and timely information that is clearly displayed will assist commanders in their C3I tactical management functions. Further, since the commander may display any subset of available tactical information, he may tailor his tactical displays to meet his particular needs.

As presented, the Generic C3I Workstation prototype maintains information concerning platform weapons status. The platform weapons status is useful in weapons-oriented C3I resource management decisions that a tactical commander must consider.

Through the automation of platform resource monitoring, information concerning mission critical resources is more readily available for displays and dissemination (via communications messages).

While the Generic C3I Workstation does not directly control the vehicular behavior of a given platform, it does provide accurate real-time platform position, course and velocity. This information may be useful to feed into an instantiation-specific platform navigation management tool. Certainly such a tool would combine own-ship position with tactical, geographic, meteorologic and oceanographic factors to derive recommended platform actions (e.g., changes in course, changes in velocity, changes in altitude or depth, bringing weapons to bear, etc.).

The Generic C3I Workstation abstract model proposes a few unique features that are not found in any other Navy C3I system (i.e., multi-network gateway service, *generic* dissimilar source information matching, common message dialogue interface, robust differentiated message archives, user defined filters and precedences, adaptable functionality, etc.). Since no one has built a system with this sort of functionality, it would be very difficult to devise a comprehensive set of system requirements at the onset. Rapid prototyping offers the systems software developer a new means of addressing hardware and software improvements. In several cases, the system prototype will be used to determine what hard-real-time requirements should be (e.g. the time required to translate messages from one format to another, the number of tracks that may be maintained by the system, etc.). In time, many of the timing constraints will become less restrictive as newer and more-capable hardware becomes available.

New tools and technology offer the fleet improved hardware performance, and the means to provide sophisticated software support tools to naval personnel. While private industry has been making strides in providing lower cost, more rugged, more dependable, and faster computers, the development of hard-real-time software systems remains difficult because there are very few tools available to help define and analyze critical system requirements.

The Generic C3I Workstation effort is an experiment in prototyping hard-real-time software systems. C3I is an excellent problem domain for the study of real-time Ada software development. Not only are C3I systems replete with timing constraints, but they also represent an operational arena within the Department of Defense where research efforts may directly contribute to improved force performance.

### 1. Lessons Learned

Most timing constraints that applied to the Generic C3I Workstation were of a very high-level nature and applied mostly to system-level response times. It is difficult to decompose a timing requirement that applies to a set of processes. This is a very important and difficult problem to resolve. Process sequences are not always clearly identified, they may vary based upon conditional parameters, and they may not be independent from one another.

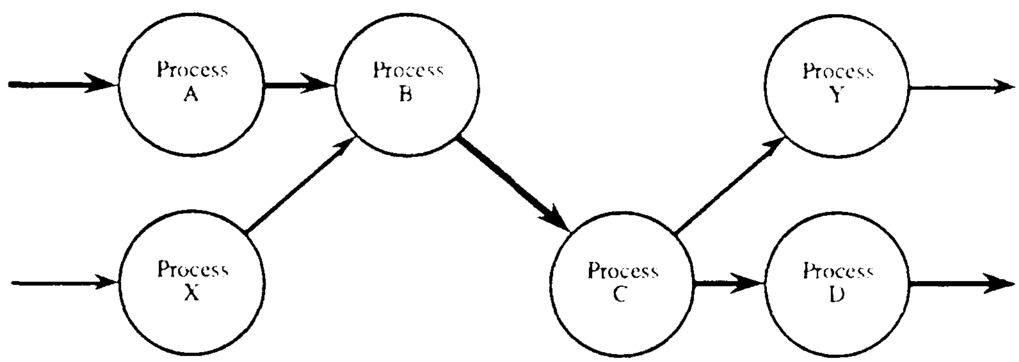



**Figure 21. Intersecting Process Sequences**

Figure 21 identifies the merging or crossing of two separate process sequences (i.e., A-B-C-D and X-B-C-Y). If during the prototyping effort, the timing constraints associated with Process B or C were to change, then both the timing sequences A-B-C-D and X-B-C-Y would need to verify that the modification did not violate their timing constraints. In large software systems, this sort of accounting becomes quickly lost.

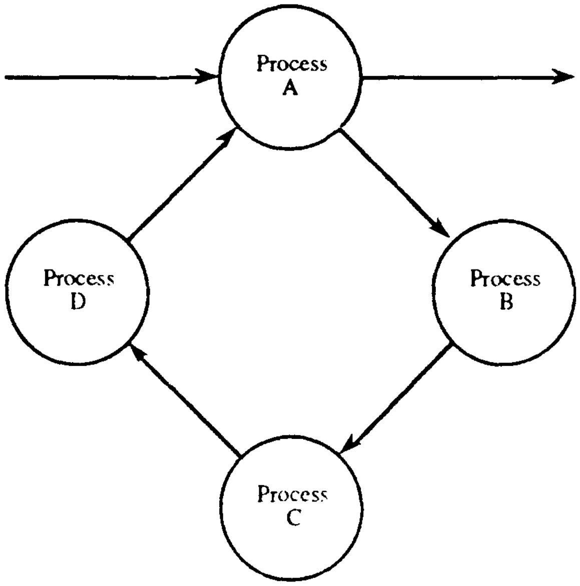



**Figure 22. Cyclic Process Sequence**

Figure 22 portrays a cyclical process sequence (analogous to a machine). If the output of Process A depends upon a subordinate process sequence, then any changes to the subordinate process sequence (A-B-C-D-A) could violate the timing constraints associated with the higher-level sequence of which Process A is a member.

### 2. Follow-on Efforts

This thesis provides an abstract model for the Generic C3I Workstation and presents a baseline functional specification for that system. This work is the first in a series of steps leading toward the rapid prototyping of a Generic C3I Workstation. At the Naval Postgraduate School, additional efforts are underway to provide a rapid prototype of a

Generic C3I Workstation (cf. LTJG Cengiz Kesoglu and LTJG Vedat Coskun entitled "Software Prototypes of C3I Stations"). LCDR Jeffrey Schweiger is developing a distributed C3I system network model with instantiated Generic C3I Workstations in a future report entitled, "Generation of a Deadlock Determination Tool for the Spec Formal Specification Language".

The groundwork has been laid for the development of a hard-real-time Ada software system in support of U.S. Navy C3I functions. Additional research, development, testing and evaluation is required to identify inherent weaknesses and areas of improvement.

## B. RECOMMENDATIONS

- An automated accounting tool for verifying software timing constraints could prove helpful in assigning system-level timing constraints to the designated sub-processes. Further, this tool could identify and resolve timing conflicts between overlapping or intersecting process sequences.

- Specific timing delays associated with naval tactical displays of varying resolution and track quantities should be ascertained. While it is believed that the performance of graphics equipment degrades in direct proportion to the amount of information being updated and displayed (either in terms of granularity, number of objects, motion, etc.), the specific timing degradations are unknown.

- Specific timing delays associated with track database functions (retrieve, add, delete) should be ascertained in support of a trade-off study to determine an optimal number of tracks to be maintained by a given track database on a given hardware implementation.

- Within the process specifications contained in Appendix D, five modules are identified that could easily be expanded in both complexity and functionality to warrant the embedding of an expert system. These modules were:

Process 1.4.1 (Inbound Message Processing) proposed an expert system for controlling network message traffic.

Process 2.1.2 (Intelligence Synthesis) proposed an expert system for identifying and correlating non-position sensor information to produce intelligence reports.

Process 3.3.5 (Identify Similarities) proposed an expert system for identifying and resolving track ambiguities.

Process 4.6.2 (Display Intel Report Screen) proposed an expert system to do the work of a Tactical Action Officer for distilling and reporting intelligence data.

Process 5.5.2 (Outbound Messages) proposed an expert system for intelligently routing outbound messages.

- Process 1.5 (Format Translator) is intended to provide the Generic C3I Workstation with the ability to take the information provided in one message and reformat the same information into a different (although similar) message format. This is a very large and difficult task that will require considerable knowledge of naval message formats and language mapping functions. It should be noted that the majority of message formats used by the U.S. Navy are classified.

- Dynamic network analysis techniques could be applied to maintaining an accurate picture of communications network participants in an ever-changing tactical environment. Communications jamming, environmental conditions, casualties and platform movements combine to make it very difficult to know at any given time who is or is not accessible by a given communications link.

- At the Naval Postgraduate School, continued efforts should be made to enhance the Computer Aided Prototyping System (CAPS). A user interface is currently being developed for CAPS. Also, the CAPS reusable software database is being improved and expanded. Yet the Generic C3I Workstation effort has underscored the need for a syntax-directed editor for generating PSDL code. Additional information is required for PSDL code that is not traditionally provided by a Yourdon software model. The transition from a Yourdon model into a model usable by CAPS is still not smooth.

As noted earlier, CAPS could benefit from a tool that automatically generates the decomposition of system-level timing requirements, as well as verifies that a change in lower-level modules does not violate system-level constraints.

## APPENDIX A

## GLOSSARY OF TERMS

The following terms are used within this thesis. These definitions come from multiple sources (including [Ref. 32]), and represent a subset of the unclassified glossary of terms from the Space and Naval Warfare Systems Command (SPAWAR) Warfare Systems Architecture and Engineering (WSA&E) Directorate's *Battle Management Architecture 2015*, Executive Summary, October 1989.

| Term | Meaning |
|---|---|
| Acquire | 1. When applied to acquisition radars, the process of detecting the presence and location of a target in sufficient detail to permit identification. 2. When applied to tracking radars, the process of positioning a radar beam so that a target is in that beam to permit the effective employment of weapons. |
| Airborne Early Warning and Control | Air surveillance and control provided by airborne early warning vehicles that are equipped with search and height-finding radar and communications equipment for controlling weapons. |
| Alternate Command Authority | One or more predesignated officers empowered by the commander through predelegation of authority to act under stipulated emergency conditions in the accomplishment of previously defined functions |
| Alternate Command Post | Any location designated by a commander to assume command post functions in the event the command post becomes inoperative. It may be partially or fully equipped and manned or it may be the command post of a subordinate unit. |
| Antiair Warfare | A U.S. Navy/U.S. Marine Corps term used to indicate that action required to destroy or reduce to an acceptable level the enemy air and missile threat. It includes such measures as the use of interceptors, bombers, anti-aircraft guns, surface-to-air and air-to-air missiles, electronic countermeasures, and destruction of the air or missile threat both before and after it is launched. Other measures which are taken to minimize the effects of hostile air action are cover, concealment, dispersion, deception (including electronic), and mobility. |

| Term | Meaning |
|---|---|
| Area Command | A command that is composed of those organized elements of one or more of the armed services, designated to operate in a specific geographical area, that are placed under a single commander, e.g.; Commander of a Unified Command, Area Commander. See also command. |
| Battle Force | A standing operational naval task force organization of carriers, surface combatants, and submarines assigned to numbered fleets. A battle force is sub-divided into battle groups. |
| Battle Group | A standing naval task group consisting of a carrier, surface combatants, and submarines as assigned in direct support, operating in mutual support with the task of destroying hostile submarines, surface and air forces within the group's assigned area of responsibility. |
| Chain of Command | The succession of commanding officers from a superior to a subordinate through which command is exercised. Also called command channel. |
| Classified Information | Official information that has been determined to require, in the interests of national security, protection against unauthorized disclosure and that has been so designated. |
| Combat Intelligence | That knowledge of the enemy, weather, and geographical features required by a commander in the planning and conduct of combat operations. (Note: NATO definition uses the words "tactical operations" in lieu of "combat operations.") |
| Command | 1. The authority that a commander in the military service lawfully exercises over subordinates by virtue of rank or assignment. Command includes the authority and the responsibility for effectively using available resources and for planning the employment of, organizing, directing, coordinating, and controlling military forces for the accomplishment of assigned missions. It also includes responsibility for health, welfare, morale, and discipline of assigned personnel. 2. An order given by a commander; that is, the will of the commander expressed for the purpose of bringing about a particular action. 3. A unit or units, an organization, or an area under the command of one individual. 4. To dominate by a field of weapon fire or by observation from a superior position. |

| Term | Meaning |
| --- | --- |
| Command and Control | The exercise of authority and direction by a properly designated commander over assigned forces in the accomplishment of the mission. Command and control functions are performed through an arrangement of personnel, equipment, communications, facilities, and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission. |
| Command and Control System | The facilities, equipment, communications, procedures, and personnel essential to a commander for planning, directing, and controlling operations of assigned forces pursuant to the mission assigned. |
| Command Center | A facility from which a commander and his representatives direct operations and control forces. It is organized to gather, process, analyze, display, and disseminate planning and operational data and perform other related tasks. |
| Command, Control and Communications (C3) | A reference to the collective activities of command and control specifically emphasizing the need for transfer of information between persons and places. |
| Command, Control, Communications and Intelligence (C3I) | A reference to the collective activities of command and control specifically emphasizing the need for transfer of information between persons and places and the intensive role of intelligence in command and control. |
| Commonality | A quality which applies to materiel or systems: 1. possessing like and interchangeable characteristics enabling each to be utilized or operated and maintained by personnel trained on the others without additional specialized training. 2. having interchangeable repair parts and/or components. 3. applying to consumable items interchangeably equivalent without adjustment. |
| Communications | A method or means of conveying information of any kind from one person or place to another. |
| Compatibility | The capability of two or more items or components of equipment or material to exist or function in the same system or environment without mutual interference. |
| Contact Report | A report indicating any detection of the enemy. |

| Term | Meaning |
|---|---|
| Control | 1. Authority which may be less than full command exercised by a commander over part of the activities of subordinate or other organizations. 2. In mapping, charting, programmetry, a collective term for a system of marks or objects on the earth or on a map or a photograph, whose positions or elevations, or both, have been or will be determined. 3. Physical or psychological pressures exerted with the intent to assure that an agent or group will respond as directed. 4. An indicator governing the distribution and use of documents, information, or material. Such indicators are the subject of intelligence community agreement and are specifically defined in appropriate regulations. |
| Coordinates | Linear or angular quantities that designate the position that a point occupies in a given reference frame or system. Also used as a general term to designate the particular kind of reference frame or system, such as plane rectangular coordinates or spherical coordinates. |
| Correlation | The relating of two or more events, reported by similar or dissimilar sources, to one another by evaluating and comparing parametrics, geographic, and time data. |
| Critical Intelligence | Intelligence that is crucial and requires the immediate attention of the commander. It is required to enable the commander to make decisions that will provide a timely and appropriate response to actions by the potential/actual enemy. It includes but is not limited to the following: 1. strong indications of the imminent outbreak of hostilities of any type (warning of attack). 2. aggression of any nature against a friendly country. 3. indications or use of nuclear-biological-chemical weapons (targets). 4. significant events within potential enemy countries that may lead to modification of nuclear strike plans. |
| Data | Representation of facts, concepts, or instructions in a formalized manner suitable for communications, interpretations, or processing by humans or by automatic means. Any representation such as characters or analog quantities to which meaning is or might be assigned. |
| Decision | In an estimate of the situation, a clear and concise statement of the line of action intended to be followed by the commander as the one most favorable to the successful accomplishment of his mission. |

| Term | Meaning |
|---|---|
| Delegation of Authority | The action by which a commander assigns part of his authority commensurate with the assigned task to a subordinate commander. While ultimate responsibility cannot be relinquished, delegation of authority carries with it the imposition of a measure of responsibility. The extent of the authority delegated must clearly be stated. |
| Detection | The discovery by any means of the presence of a person, object, or phenomenon of potential military significance. |
| Dissimilar Source Integration | The integration of data or information from diverse sources, including radar, IFF, EW, acoustic, visual and/or a variety of other sensor inputs. |
| Duplex | A full duplex circuit provides two channels or frequencies linking two different stations, allowing the simultaneous exchange of information. |
| Electronic Warfare Support Measures (ESM) | That division of electronic warfare involving actions taken under direct control of an operational commander to search for, intercept, identify, and locate sources of radiated electromagnetic energy for the purpose of immediate threat recognition. |
| Emission Control (EMCON) | The selective and controlled use of electromagnetic, acoustic, or other emitters: 1. to optimize command and control capabilities while minimizing, for operations security (OPSEC), detection by enemy sensors. 2. to minimize mutual interference among friendly systems. 3. to execute a military deception plan. |
| Encrypt | To convert plain text into unintelligible form by means of a cryptosystem. (Note: The term encrypt covers the meanings of encipher and encode.) |
| Essential Communications Traffic | Transmission (record/voice) of any precedence which must be sent electronically in order for the command or activity concerned to avoid serious impact on mission accomplishment or safety or life. |
| Essential Elements of Information | The critical items of information regarding the enemy and the environment needed by the commander by a particular time to relate with other available information and intelligence in order to assist in reaching a logical decision. |

| Term | Meaning |
|---|---|
| Fleet | An organization of ships, aircraft, marine forces, and shore-based activities under the command of commander or commander in chief who may exercise operational as well as administrative control. |
| Force | 1. An aggregation of military personnel, weapon systems, vehicles, and necessary support, or combination thereof. 2. A major subdivision of a fleet. |
| Identification | The process of determining the friendly or hostile character of an unknown detected contact. |
| Identify | To affix a label within the classification taxonomy to an entity (target). |
| Imagery Intelligence (IMINT) | Intelligence derived from the exploitation of information collected by visual photography, infrared sensors, lasers, electro-optics and radar sensors such as synthetic aperture radar wherein images of objects are reproduced optically or electronically on film, electronic display devices or other media. |
| Information (Intelligence) | Unevaluated material of every description, including that derived from observations, reports, rumors, imagery, and other sources that, when processed, may produce intelligence data. |
| Intelligence | The product resulting from the collection , processing, integration, analysis, evaluation and interpretation of available information concerning foreign countries or activities. |
| Interchangeability | A condition which exists when two or more items possess such functional and physical characteristics as to be equivalent in performance and durability, and are capable of being exchanged one for the other without alteration of the items themselves or of adjoining items, except for adjustment, and without selection for fit and performance. |
| Interface | A boundary or point common to two or more similar or dissimilar command and control systems, subsystems, or other entities against which or at which necessary information flow takes place. |

| Term | Meaning |
|---|---|
| Interior Communications | Rapid communications facilities (electrical, acoustical, or mechanical) interconnecting the various operational spaces of a naval ship, aircraft, or other activities. |
| Interoperability | 1. The ability of systems, units or forces to provide services to and accept services from other systems, units or forces and to use the services so exchanged to enable them to operate effectively together. 2. The condition achieved among communications/electronics equipment when information orservices can be exchanged directly and satisfactorily between them and/or their users. The degree of interoperability should be defined when referring to the specific cases. |
| Joint Operational Tactical System (JOTS) | JOTS is the U.S. Navy developmental prototype system for tactical decision support. JOTS is a battle management systemfor use at sea by battle force and battle group command staffs and on shore by command center staffs. |
| Joint Operations | Operations carried out by elements of two or more services of the Department of Defense. |
| Link (Communications) | A general term used to indicate the existence of communications facilities between two points. |
| Logistics | The science of planning and carrying out the movement andmaintenance of forces. In its most comprehensive sense,those aspects of military operations which deal with: 1. design and development, acquisition, storage, movement, distribution, maintenance, evacuation, and disposition of materiel. 2. movement, evacuation, and hospitalization of personnel. 3. acquisition or construction, maintenance, operation, and disposition of facilities. 4. acquisition or furnishing of services. |
| Management | A process of establishing and attaining objectives to carry out responsibilities. Management consists of those continuing actions of planning, organizing, directing, coordinating, controlling, and evaluating the use of men, money, materials, and facilities to accomplish missions and tasks. Management is inherent in command, but it does not include as extensive authority and responsibility as command. |

| Term | Meaning |
|---|---|
| Message | Any though or idea expressed briefly in plain or secret language, prepared in a form suitable for transmission by any means of communication. |
| Mission | 1. The task, together with the purpose, which clearly indicates the action to be taken and the reason therefor. 2. In common usage, especially when applied to lower military units, a duty assigned to an individual or unit; a task. 3. The dispatching of one or more aircraft to accomplish one particular task. |
| National Command Authorities (NCA) | The President and the Secretary of Defense or their duly deputized alternates or successors. |
| Naval Tactical Data System (NTDS) | A complex of data inputs, user consoles, converters, adapters, and radio terminals interconnected with high-speed general purpose computers and its stored programs. Combat data is collected, processed, and composed into a picture of the overall tactical situation which enables the force commander to make rapid, accurate evaluations and decisions. |
| Net (Communications) | An organization of stations capable of direct communications on a common channel or frequency. |
| Officer in Tactical Command | In maritime usage, the senior officer present eligible to assume command, or the officer to whom he has delegated tactical command. |
| Order | A communication, written, oral, or by signal, that conveys instructions from a superior to a subordinate. In a broad sense, the term "order" and "command" are synonymous. However, an order implies discretion as to the details of execution, whereas a command does not. |
| Order of Battle | The identification, strength, command structure, and disposition of the personnel, units, and equipment of any military force. |
| Passive | In surveillance, an adjective applied to actions or equipments that emit no energy capable of being detected. |
| Periodic Intelligence Summary (PERINTSUM) | A report of the intelligence situation in a tactical operation, normally produced at the corps level or its equivalent, and higher, usually at intervals of 24 hours, or as directed by the commander. |

| Term | Meaning |
|---|---|
| Possible | A term used to qualify a statement made under conditions wherein some evidence exists to support the statement. This evidence is sufficient to warrant mention, but insufficient to warrant assumption as true. See also probable. |
| Probable | A term used to qualify a statement made under conditions wherein the available evidence indicates that the statement is factual until there is further evidence in confirmation or denial. See also possible. |
| Reaction Time | 1. The elapsed time between the initiation of an action and the required response. 2. The time required between the receipt of an order directing an operation and the arrival of the initial element of the force concerned in the designated area. |
| Real-time | 1. The absence of delay, except for the time required for the transmission by electromagnetic energy, between the occurrence of an event or the transmission by electromagnetic energy, between the the occurrence of an event or the transmission of data, and the knowledge of the event, or reception of the data at some other location. 2. A real-time event or data transfer is one which must be accomplished within an allotted amount of time or the accomplishment of the action has either no or diminishing value. |
| Record Information | All forms (e.g., narrative, graphic, data, computer memory) of information registered in either temporary or permanent for so that it can be retrieved, reproduced, or preserved. |
| Resolution distinguished | A measurement of the smallest detail that can be by a sensor system under specific conditions. |
| Responsiveness | The ability of a system or component to provide a desired level of service within the time envelope prescribed for the urgency level of the information being transmitted or processed. It measures writer-to-reader time, but does not include thought or composition processes. |
| Rules of Engagement (ROE) | Directives issued by competent military authority which delineate the circumstances and limitations under which United States armed forces will initiate and/or continue combat engagement with other forces encountered. |

| Term | Meaning |
|---|---|
| Sensor | Equipment that detects, and may indicate, and /or record objects and activities by means of energy or particles emitted, reflected, or modified by objects. |
| Signature | The characteristic pattern of the target provided by detection and identification equipment. |
| Standard | An exact value, a physical entity, or an abstract concept, established and defined by authority, custom, or common consent to serve as a reference, model, or rule in measuring quantities or qualities, establishing practices or procedures, or evaluating results. A fixed quantity or quality. |
| Standardization | The process by which the Department of Defense achieves the closest practicable cooperation among the armed services and defense agencies for the most efficient use of research, development, and production resources, and agrees to adopt on the broadest possible basis the use of: 1. common or compatible operational, administrative, and logistic procedures. 2. common or compatible technical procedures and criteria. 3. common, compatible, or interchangeable supplies, components, weapons, or equipment. 4. common or compatible tactical doctrine with corresponding organizational compatibility. |
| Tactical Digital Information Link (TADIL) | A communications link which uses standardized message formats and transmission characteristics for specific equipment. |
| Tactics | 1. The employment of units in combat. 2. The ordered arrangement and maneuver of units in relation to each other and/or the enemy in order to maximize their effectiveness. |
| TADIL "A" | A netted digital data link using parallel transmission frame characteristics and standard message formats at either 2250 or 1364 bits per second. Also referred to as Link 11. |
| TADIL "B" | A point-to-point digital data link using serial transmissions frame characteristics and standard message formats at a basic speed of 1200 bits per second. This data link interconnects tactical air defense and aircraft control units of the implementing services. |

| Term | Meaning |
|---|---|
| TADIL "C" | A time division digital data transmission link between control station and controlled aircraft. It provides the capability for automatic transmissions or orders, status, and other information. Data exchange is accomplished on a fully automatic link at 5000 bits per second, using serial transmission. Also referred to as Link 4A. |
| Target Acquisition | The detection, identification, and location of a target in sufficient detail to permit the effective employment of weapons. |
| Target Analysis | An examination of potential targets to determine military importance, priority of attack, and weapons required to obtain a desired level of damage or causalities. |
| Target Discrimination | The ability of a surveillance or guidance system to identify or engage any one target when multiple targets are present. |
| Target Resolution | The minimum difference in bearing, range, or elevation between two targets that will allow obtaining data on either target. |
| Targeting | The process of selecting targets and matching the appropriate response to them, taking into account operational requirements and capabilities. |
| Task Force | 1. A temporary grouping of units, under one commander, formed for the purpose of carrying out a specific operation or mission. 2. Semi-permanent organization or units, under one commander, formed for the purpose of carrying out a continuing specific task. 3. A component of a fleet organized by the commander of a task fleet or higher authority for the accomplishment of a specific task or tasks. |
| Task Group | A component of a naval task force organized by the commander of the task force or higher authority. |
| Theater | The geographic area outside the continental United States for which a commander of a unified or specified command has been assigned military responsibility. |

| Term | Meaning |
|---|---|
| Track | 1. A series of related contacts displayed on a plotting board. 2. To display or record the successive positions of a moving object. 3. To lock onto a point of radiation and obtain guidance therefrom. 4. To keep a gun properly aimed, or to point continuously at a moving target. 5. The actual path of an aircraft above, or a ship on, the surface of the earth. The course is the path that is planned; the track is the path actually taken. |
| Track Correlation | Correlating track information for identification purposes using all available data. |
| Track Telling | The process of communicating air surveillance and tactical data information between command and control systems or between facilities within systems. Telling may be classified into the following types:<br>1. Back tell -- The transfer of information from a higher to a lower echelon of command.<br>2. Cross tell, or Lateral tell -- The transfer of information between facilities at the same operational level of command.<br>3. Forward tell -- The transfer of information to a higher level of command.<br>4. Overlap tell -- The transfer of information to an adjacent facility's area of responsibility.<br>5. Relateral tell -- The relay of information between facilities through the uses of a third facility. This type of telling is appropriate between automated facilities in a degraded communications environment. |
| Weapon | An assembled and ready for delivery conventional or nuclear device in the military configuration. For naval gunfire or artillery, a weapon is a complete round; for a rocket, the motor plus the warhead; for a missile, the complete missile to include the warhead; for a torpedo, the complete torpedo to include the warhead; for air-delivered weapons, the warhead in the bomb; and for an atomic demolition munition, the complete munition. |
| Weapon System | A weapon and those components required for its operation. (The term is not precise unless specific parameters are established.) |

# APPENDIX B

# INITIAL GENERIC C3I WORKSTATION FUNCTIONAL SPECIFICATION

```
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--    gc3iws
--.TITLE.
--    generic command, control, communications and intelligence
--    workstation
--.SYNOPSIS.
--    MACHINE    gc3iws
--.DESCRIPTION.
--    This SPEC module encapsulates the abstract functional
--    specification for a computer system which can be used as a general
--    purpose command, control, communications and intelligence
--    workstation for Navy battlegroup operations.  The entire software
--    system is modularly built and is composed of those pieces
--    specified in the INHERIT clauses.
--.AUTHOR.
--    Jeff Schweiger
--.SUPPLEMENTARY.
--    CS4530
--.VERSION.
--    gc3iws.spec 1.3
--.DATE.
--    19 September 1990
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

MACHINE gc3iws
  INHERIT communications_interface
  INHERIT sensor_interface
  -- interface to platform sensors and navigation system
  INHERIT track_database_manager
  INHERIT track_data_display
  -- interface to track manager
  INHERIT tactical_command_display
  -- interface to battle manager and local area network
  INHERIT weapons_systems_interface
  INHERIT message_archives
  INHERIT normalization
  INHERIT navigation_interface

  STATE
  INVARIANT true
  INITIALLY true
END
```

```
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--    comms_interface
--.TITLE.
--    generic command, control, communications and intelligence
--    workstation communications interface
--.SYNOPSIS.
--    MACHINE comms_interface
--.DESCRIPTION.
--    This SPEC module encapsulates the abstract functional
--    specification for the communications interface for a general
--    purpose command, control, communications and intelligence
--    workstation for Navy battlegroup operations.
--.AUTHOR.
--    Jeff Schweiger
--.SUPPLEMENTARY.
--    CS4910
--.VERSION.
--    commsint.spec     1.2
--.DATE.
--    23 September 1990
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

MACHINE comms_interface
  INHERIT interface_definitions

  STATE (archive_set: set{archive_id},
         emcon_status: emcon_message,
         network_set: set{network_setup_tuple},
         message_queue: set{message})
  INVARIANT true
  INITIALLY archive_set = {all},
            emcon_status = unrestricted,
            network_set = {},
            message_queue = {}

  MESSAGE text_message (m: message)
     SEND electronic_mail (m: message) TO tactical_command_display
     SEND (m: message) TO message_archive

  MESSAGE track_message (t: track)
     SEND add_track_tuple (a: change_track_msg)
          TO track_database_manager
      WHERE a.origin = t.origin,
            a.change = add,
            a.track = t

  MESSAGE transmit_command (m: message)
     WHEN emcon_status = emcon
     TRANSITION message_queue = *message_queue U {m}
     OTHERWISE
      SEND (m: message) TO external_communications_link
```

```
MESSAGE network_setup (n: network_setup_tuple)
   TRANSITION network_set = *network_set U {n}

MESSAGE emissions_control_command (e: emcon_message)
   TRANSITION emcon_status = e

MESSAGE archive_setup (a: archive_setup_message)
   WHEN a = {all}
    TRANSITION archive_set = {all}
   WHEN a = {ownship}
    TRANSITION archive_set = {ownship}
   WHEN a ~= {all} & archive_set ~= {all}
    TRANSITION archive_set = *archive_set U {a}
   OTHERWISE
    TRANSITION archive_set = {a}

MESSAGE initiate_transmit {t: type} (attr: function{track, t},
                          op: function{t, t, boolean}, value: t)
   SEND database_request(attr, op, value) TO track_database_manager

MESSAGE (t: track)
   SEND (m: message) TO external_communications_link
    WHERE m.message_body = t

END
```

```
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--    sensor_interface
--.TITLE.
--    generic command, control, communications and intelligence
--    workstation sensor interface
--.SYNOPSIS.
--    FUNCTION sensor_interface
--.DESCRIPTION.
--    This SPEC module encapsulates the abstract functional
--    specification for the sensor interface for a general purpose
--    command, control, communications and intelligence workstation
--    for Navy battlegroup operations.
--.AUTHOR.
--    Jeff Schweiger
--.SUPPLEMENTARY.
--    CS4910
--.VERSION.
--    sensorint.spec    1.2
--.DATE.
--    19 September 1990
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

FUNCTION sensor_interface
  INHERIT interface_definitions

  MESSAGE sensor_contact_report (c: track)
     SEND normalize_contact (c: track) TO normalization

  MESSAGE normalized_contact_report (n: track)
     SEND add_track_tuple (a: change_track_msg)
          TO track_database_manager
      WHERE
         a.origin = n.origin
         a.change = add
         a.track = n.track

  MESSAGE sensor_intell_report (i: intell_report_msg)
     SEND intelligence_report (i: intell_report_msg)
          TO track_data_display

END
```

```
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--    track_database_manager
--.TITLE.
--    generic command, control, communications and intelligence
--    workstation track database manager module
--.SYNOPSIS.
--    MACHINE     track_database_manager
--.DESCRIPTION.
--    This SPEC module encapsulates the abstract fu  tional
--    specification for the track database manager ior a general purpose
--    command, control, communications and intelligence workstation for
--    Navy battlegroup operations.
--.AUTHOR.
--    Jeff Schweiger
--.SUPPLEMENTARY.
--    CS4910
--.VERSION.
--    trackdbm.spec     1.3
--.DATE.
--      23 September 1990
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

MACHINE     track_database_manager

  INHERIT interface_definitions
  STATE(track_data_base: set{track}, max_num_tracks: integer,
        desired_classes: set{class_range_tuple},
        archive_timeout: integer, monitor_range: integer,
        monitor_mode: mode_type, refresh_rate: integer)
  INVARIANT true
  INITIALLY track_data_base = {(t.id = ownship_id,
                                t.origin = ownship_id,
                                t.class = ownship, t.latitude = 0,
                                t.longitude = 0, t.depth = 0,
                                t.course = 0, t.velocity = 0,
                                t.time = 0, t.textfield = "")},
            max_num_tracks = 1000,
            desired_classes = {(air,1000)},
            archive_timeout = 60,  -- units in minutes
            monitor_range = 1024,  -- units in nautical miles
            monitor_mode = off,
            refresh_rate = 60 -- units in seconds

  MESSAGE ownship_posit_report (o: track)
     CHOOSE (t: track SUCH THAT t IN *track_data_base & t.id = ownship)
     TRANSITION track_data_base = *track_data_base - {t} U {o}

  MESSAGE add_track_tuple (a: change_track_msg)
     TRANSITION track_data_base = *track_data_base U {a.track}
```

```
MESSAGE delete_track_tuple (d: change_track_msg)
   WHEN d.track IN track_data_base
    CHOOSE (t: track SUCH THAT t IN *track_data_base &
            t.id = d.track.id)
         TRANSITION track_data_base = *track_data_base - {t}
   OTHERWISE
    REPLY (s: string)
       WHERE s = "no such track"

MESSAGE update_track_tuple (u: change_track_msg)
   WHEN u.track IN track_data_base
    CHOOSE (t: track SUCH THAT t IN *track_data_base &
            t.id = u.track.id)
         TRANSITION track_data_base = *track_data_base - {t} U
                                      {u.track}
   OTHERWISE
    REPLY (s: string)
       WHERE s = "no such track"

MESSAGE database_request {t: type} (attr: function{track, t},
                          op: function{t, t, boolean}, value: t)
   REPLY (s: set{track})
    WHERE s = {tr: track :: op(attr(tr), value)}

MESSAGE track_ambiguity (a: ambiguity_report)
    -- track_ambiguity message originates within a track database
    -- monitor expert system which does track correlation/data fusion.
    -- This system will be defined/specified at a future time.
   SEND ambiguity_resolution_notice (a: ambiguity_report) TO
       track_data_display

MESSAGE set_track_filter (m: integer, d: class_range_tuple)
   TRANSITION max_num_tracks = m,
              desired_classes = d

MESSAGE set_monitor_constraints (a: integer, mr: integer,
                                 mm: mode_type, r: integer)
   TRANSITION archive_timeout = a,
             monitor_range = mr,
             monitor_mode = mm,
             refresh_rate = r

END
```

```
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--    track_data_display
--.TITLE.
--    generic command, control, communications and intelligence
--    workstation track data display module
--.SYNOPSIS.
--    FUNCTION track_data_display
--.DESCRIPTION.
--    This SPEC module encapsulates the abstract functional
--    specification for the track data display for a general purpose
--    command, control, communications and intelligence workstation
--    for Navy battlegroup operations.
--.AUTHOR.
--    Jeff Schweiger
--.SUPPLEMENTARY.
--    CS4910
--.VERSION.
--    trackdisp.spec    1.4
--.DATE.
--    23 September 1990
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

FUNCTION track_data_display
  INHERIT interface_definitions

  MESSAGE user_view_track_request {t: type} (attr: function{track, t},
                                   op: function{t, t, boolean}, value: t)
     SEND database_request (attr, op, value) TO track_database_manager

  MESSAGE (t: track)
     SEND (t: track) TO user

  MESSAGE user_add_track_request (t: track)
     SEND add_track_tuple (a: change_track_msg)
          TO track_database_manager
      WHERE a.origin = user,
            a.change = add,
            a.track = t

  MESSAGE user_delete_track_request (t: track)
     SEND delete_track_tuple (d: change_track_msg)
          TO track_database_manager
      WHERE d.origin = t.origin,
            d.change = delete,
            d.track = t

  MESSAGE user_update_track_request (t: track)
     SEND update_track_tuple (u: change_track_msg)
          TO track_database_manager
      WHERE u.origin = user,
            u.change = update,
            u.track = t

  MESSAGE ambiguity_resolution_notice (a: ambiguity_report)
     SEND resolution_notice (a: ambiguity_report) TO user
```

```
MESSAGE intelligence_report (i: intell_report_msg)
   SEND intel_report (i: intell_report_msg) TO user

MESSAGE user_archive_setup_message (a: archive_setup_message)
   SEND archive_setup (a: archive_setup_message)
        TO communications_interface

MESSAGE user_initiate_transmit (t: type) (attr: function(track, t),
                              op: function(t, t, boolean), value: t)
   SEND initiate_transmit (attr, op, value)
        TO communications_interface

MESSAGE user_set_track_filter (m: integer, d: class_range_tuple)
   SEND set_track_filter (m: integer, d: class_range_tuple )
        TO track_database_manager

MESSAGE user_set_monitor_constraints (a: integer, mr: integer,
                                   mm: mode_type, r: integer)
   SEND set_monitor_constraints (a: integer, mr: integer,
                               mm: mode_type, r: integer)
        TO track_database_manager

END
```

```
--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--    tactical_command_display
--.TITLE.
--    generic command, control, communications and intelligence
--    workstation tactical command display
--.SYNOPSIS.
--    FUNCTION tactical_command_display
--.DESCRIPTION.
--    This SPEC module encapsulates the abstract functional
--    specification for the tactical command display for a general
--    purpose command, control, communications and intelligence
--    workstation for Navy battlegroup operations.
--.AUTHOR.
--    Jeff Schweiger
--.SUPPLEMENTARY.
--    CS4910
--.VERSION.
--    taccomdisp.spec        1.4
--.DATE.
--    23 September 1990
--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

FUNCTION tactical_command_display
  INHERIT interface_definitions

  MESSAGE electronic_mail (m: message)
     SEND (s: string) TO user
      WHERE s = "Electronic Mail Received."

  MESSAGE emergency_weapon_status_report (e: status_report_msg)
     SEND (w: weapon_report) TO user
      WHERE w.alert_flag = emergency
            w.origin = e.origin
            w.current_status = e.current_status
            w.load_out = e.load_out
            w.text_string = e.text_string

  MESSAGE user_weapon_status_request (i: weapon_id)
     SEND status_query (i: weapon_id) TO weapons_system_interface

  MESSAGE status_report (s: status_report_msg)
     SEND (w: weapon_report) TO user
      WHERE w.alert_flag = routine
            w.origin = s.origin
            w.current_status = s.current_status
            w.load_out = s.load_out
            w.text_string = s.text_string

  MESSAGE new_message_from_user (m: message)
     SEND transmit_command (m: message) TO communications_interface
```

```
   MESSAGE user_view_track_request (t: type) (attr: function(track, t),
                                    op: function(t, t, boolean), value: t)
      SEND database_request (attr, op, value) TO track_database_manager

   MESSAGE (t: track)
      SEND (t: track) TO user

   MESSAGE user_read_msg_request (h: header)
      SEND retrieve_message (h: header) TO message_archive

   MESSAGE message_from_archive (m: message)
      SEND (m: message) TO user

   MESSAGE user_emcon_command (e: emcon_message)
      SEND emissions_control_command (e: emcon_message)
           TO communications_interface

   MESSAGE network_setup_input (n: network_setup_tuple)
      SEND network_setup (n: network_setup_tuple)
           TO communications_interface

END
```

```
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--    weapons_systems_interface
--.TITLE.
--    generic command, control, communications and intelligence
--    workstation weapons systems interface module
--.SYNOPSIS.
--    MACHINE     weapons_systems_interface
--.DESCRIPTION.
--    This SPEC module encapsulates the abstract functional
--    specification for the weapons systems interface for a general
--    purpose command, control, communications and intelligence
--    workstation for Navy battlegroup operations.
--.AUTHOR.
--    Jeff Schweiger
--.SUPPLEMENTARY.
--    CS4910
--.VERSION.
--    wepnsint.spec      1.3
--.DATE.
--    18 September 1990
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

MACHINE     weapons_systems_interface
  INHERIT interface_definitions

  STATE (ws: map{origin, weapon_status})
  INVARIANT true
  INITIALLY ALL(id: origin :: ws[id].current_status = secured,
                              ws[id].load_out = 0)

  MESSAGE weapon_status_update (w: weapon_status)
     TRANSITION ws = bind (w.origin, w, *ws)
     WHEN w.current_status = damaged | out_of_ammunition
      SEND emergency_weapon_status_report (e: status_report_msg)
            TO tactical_command_display
      WHERE e.origin = w.origin,
            e.current_status = w.current_status,
            e.load_out = w.load_out,
            e.text_string = "Weapon is inoperative!!"
     OTHERWISE
      REPLY nil
             -- no op

  MESSAGE status_query (i: weapon_id)
     WHEN i IN ws
       REPLY status_report (s: status_report_msg)
         WHERE s.origin = i,
               s.current_status = ws[i].current_status,
               s.load_out = ws[i].load_out,
               s.text_string = "Normal report"
     OTHERWISE
      REPLY (st: string)
         WHERE st = "No such weapon onboard."

END
```

```
--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--      message_archives
--.TITLE.
--      generic command, control, communications and intelligence
--      workstation message archive
--.SYNOPSIS.
--      MACHINE     navigation_interface
--.DESCRIPTION.
--      This SPEC module encapsulates the abstract functional
--      specification for a electronic mail textfile data store for a
--      general purpose command,control, communications and intelligence
--      workstation for Navy battlegroup operations.  The message archive
--      stores incoming messages and sends them to other modules upon
--      request
--.AUTHOR.
--      Jeff Schweiger
--.SUPPLEMENTARY.
--      CS4910
--.VERSION.
--      archive.spec      1.1
--.DATE.
--      22 September 1990
--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

MACHINE message_archives
  INHERIT interface_definitions
  STATE(messages: set(m: message))
  INVARIANT true
  INITIALLY messages = {},
          archive_ids = {(ALL)}

  MESSAGE receive_message (r: message)
     TRANSITION messages = *messages U {r}

  MESSAGE user_read_msg_request (h: header)
     REPLY message_from_archive (a: message)
      WHERE a.header = h

END
```

```
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--    normalization
--.TITLE.
--    generic command, control, communications and intelligence
--    workstation contact normalization function
--.SYNOPSIS.
--    FUNCTION sensor_interface
--.DESCRIPTION.
--    This SPEC module encapsulates the abstract functional
--    specification for a function that takes relative positioning
--    information from a sensor and returns normalized contact data
--    based on ownship position data.  This function is called from the
--    sensor interface module for a general purpose command, control,
--    communications and intelligence workstation for Navy battlegroup
--    operations.
--.AUTHOR.
--    Jeff Schweiger
--.SUPPLEMENTARY.
--    CS4910
--.VERSION.
--    normalization.spec      1.1
--.DATE.
--    23 September 1990
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

FUNCTION normalization
  INHERIT interface_definitions

  MESSAGE normalize_contact (c: track)
     SEND request_ownship_position (t: track) TO navigation_interface

  MESSAGE (t: track, ownship_position: nav_data)
     SEND normalized_contact_report (n: track) TO sensor_interface
      WHERE n = normalize(t, ownship_position)

  CONCEPT normalize (t: track, c: nav_data) VALUE (n: track)
     WHERE ?
     -- Normalize is not fully defined.  It indicates that the raw
     -- contact data from the sensor is adjusted based on ownship
     -- position at the time of the contact report

END
```

```
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--    navigation_interface
--.TITLE.
--    generic command, control, communications and intelligence
--    workstation navigation system interface
--.SYNOPSIS.
--    MACHINE     navigation_interface
--.DESCRIPTION.
--    This SPEC module encapsulates the abstract functional
--    specification for the navigation interface for a general purpose
--    command, control, communications and intelligence workstation for
--    Navy battlegroup operations.  This module 'wraps' the external
--    navigation system and keeps track of ownship position
--.AUTHOR.
--    Jeff Schweiger
--.SUPPLEMENTARY.
--    CS4910
--.VERSION.
--    navint.spec 1.1
--.DATE.
--    22 September 1990
--+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

MACHINE navigation_interface
  INHERIT interface_definitions
  STATE(s: set{own_ship_navigation_information: nav_data})
  INVARIANT true
  INITIALLY s = {(own_ship_navigation_information.course = 0,
                  own_ship_navigation_information.velocity = 0,
                  own_ship_navigation_information.latitude = 0,
                  own_ship_navigation_information.longitude = 0,
                  own_ship_navigation_information.depth = 0,
                  own_ship_navigation_information.time = 0)}
    -- Own-ship position initializes to zero values

  MESSAGE receive_nav_data (n: nav_data)
     TRANSITION s = *s U {n}
     SEND ownship_posit_report (o: track)
      WHERE o.id = ownship_id,
            o.track_origin = ownship_id,
            o.class = ownship,
            o.latitude = n.latitude,
            o.longitude = n.longitude,
            o.depth = n.depth,
            o.course = n.course,
            o.velocity = n.velocity,
            o.time = n.time,
            o.textfield = ""

  MESSAGE request_ownship_position (t: track)
     REPLY ( t: track, o: nav_data)
      WHERE o.time = t.time

END
```

```
--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--.NAME.
--    interface_definitions
--.TITLE.
--    generic command, control, communications and intelligence
--    workstation interface definitions
--.SYNOPSIS.
--    DEFINITION interface_definitions
--.DESCRIPTION.
--    This SPEC module encapsulates concepts used to catagorize
--    communications messages for the communications interface for
--    a general purpose command, control, communications and
--    intelligence workstation for Navy battlegroup operations.
--.AUTHOR.
--    Jeff Schweiger
--.SUPPLEMENTARY.
--    CS4910
--.VERSION.
--    commsdef.spec    1.1
--.DATE.
--    23 September 1990
--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

DEFINITION interface_definitions

  CONCEPT archive_id: type
    WHERE archive_id = special_id | link_id

  CONCEPT special_id: type
     WHERE special_id = enumeration{
        all,
        ownship
     }

  CONCEPT link_id: type
     WHERE link_id = tuple{
        link_desig :: link_type,
        channel_id :: string
     }

  CONCEPT link_type: type
     WHERE link_type = enumeration{
        link4a,
        link11,
        link16,
        otcixs              -- others can be inserted as appropriate
     }

  CONCEPT emcon_message: type
     WHERE emcon_message = enumeration{
        emcon,
        restricted,
        unrestricted
     }
```

```
CONCEPT network_setup_tuple: type
   WHERE network_setup_tuple = tuple{
      net_link_id :: link_id,
      addressee :: string,
      net_control_flag :: net_unit_type,
      via_line :: string
   }

CONCEPT net_unit_type: type
   WHERE net_unit_type = enumeration{
      m,                   --  represents master unit
      p,                   --  participating unit
      a                    --  alternate master unit
   }

CONCEPT message: type
   WHERE message = tuple {
      header :: header,
      message_body :: string,
      routing_line :: string
   }

CONCEPT header: type
   WHERE header = tuple {
      classification :: string,
      precedence :: string,
      sender :: string,
      addressee :: string,
      via_line :: string,
      info_line :: string,
      subj_line :: string
   }

CONCEPT track: type
   WHERE track = tuple{
      id :: string,
      origin :: string,
      time :: time,
      class :: class_type,
      iff_class :: iff_class_type,
      latitude :: integer,
      longitude :: integer,
      depth :: integer,
      course :: integer,
      velocity :: integer,
      textfield :: string
   }

CONCEPT class_type: type
   WHERE class_type = enumeration{
      surface,
      subsurface,
      air
   }
```

```
CONCEPT iff_class_type: type
   WHERE iff_class_type = enumeration{
      friendly,
      hostile,
      neutral,
      unknown
   }                           -- others as desired

CONCEPT change_track_msg: type
   WHERE change_track_msg = tuple{
      origin :: string,
      change :: change_type,
      track :: track
   }

CONCEPT change_type: type
   WHERE change_type = enumeration{
      add,
      delete,
      update
   }

CONCEPT intell_report_msg: type
   WHERE intell_report_msg = tuple{
      origin :: string,
      intelligence_data :: string
   }

CONCEPT class_range_tuple: type
   WHERE class_range_tuple = tuple{
      track_class :: class_type,
      range :: integer
   }

CONCEPT mode_type: type          -- used with track database monitor
   WHERE mode_type = enumeration{
      automatic,
      advise,
      off
   }

CONCEPT amibiguity_report: type
   WHERE ? -- this report remains to be defined but is used to inform
         -- the operator of track ambiguities

CONCEPT archive_setup_message: type
   WHERE archive_setup_message = set {archive_id}

CONCEPT status_report_message: type
   WHERE status_report_message = tuple{
      origin :: string,
      current_status :: weapon_status,
      load_out :: string,
      text_string :: string
   }
```

```
CONCEPT weapon_status: type
   WHERE weapon_status = enumeration{
      damaged,
      reloading,
      launching,
      ready,
      service_required,
      slewing,
      out_of_ammunition,
      secured,
      maintenance,
      engaging
   }

 CONCEPT weapon_id: type
    WHERE weapon_id = string

 CONCEPT nav_data: type
    WHERE nav_data = tuple{
      time :: time,
      latitude :: integer,
      longitude :: integer,
      depth :: integer,
      course :: integer,
      velocity :: integer
    }

END
```

# APPENDIX C

## GENERIC C3I WORKSTATION FUNCTIONAL DECOMPOSITION

This section utilizes the Yourdon software modeling approach [Ref. 26] for functionally decomposing the Generic C3I Workstation. Following the listing of the module hierarchy, a corresponding set of data flow diagrams is provided. The process specifications for the lowest level modules are included in Appendix D. Appendix E is the data dictionary that corresponds with the data flow diagrams.

### 1 COMMUNICATIONS INTERFACE (ACCEPT, FORMAT & ROUTE)

1.1 COMMS MESSAGE CONTROL

- 1.1.1 MESSAGE FORWARDING
- 1.1.2 INPUT MESSAGE RECEIVER
- 1.1.3 LINK-4A MESSAGE CONTROL
- 1.1.4 LINK-11 MESSAGE CONTROL
- 1.1.5 LINK-16 MESSAGE CONTROL
- 1.1.6 OTCIXS MESSAGE CONTROL

1.2 MESSAGE LIBRARIAN

- 1.2.1 ARCHIVE FILTER
- 1.2.2 SAVE MESSAGE TEXT
- 1.2.3 TRACK-TEXT SORTER

1.3 TRACK EXTRACTOR

- 1.3.1 TRACK-CONTACT SORTER
- 1.3.2 COMMS TRACK SYNTHESIS
- 1.3.3 COMMS CONTACT SYNTHESIS
- 1.3.4 TUPLE FORWARDING

1.4 COMMUNICATIONS NETWORK MONITOR & CONTROL

- 1.4.1 INBOUND MESSAGE PROCESSING
- 1.4.2 OUTBOUND MESSAGE PROCESSING
- 1.4.3 TRANSLATION INTERFACE
- 1.4.4 TRANSMISSION FORWARDING

1.5 FORMAT TRANSLATOR

1.6 PERIODIC TRANSMISSION GENERATOR

- 1.6.1 PERIODIC REPORT GENERATOR
- 1.6.2 TRACK REPORT
- 1.6.3 MESSAGE FORMAT TEMPLATE

## 2 SENSOR INTERFACE (ACCEPT & FORMAT)

### 2.1 SENSOR INTERFACE CONTROL

2.1.1 SENSOR CONTACT SYNTHESIS
2.1.2 INTELLIGENCE SYNTHESIS
2.1.3 RADAR INTERFACE CONTROL
2.1.4 SONAR INTERFACE CONTROL
2.1.5 INFRARED INTERFACE CONTROL
2.1.6 ESM INTERFACE CONTROL

### 2.2 SENSOR INFORMATION NORMALIZATION

### 2.3 OWNSHIP LOCATION MONITOR

### 2.4 RELATIVE-TO-ABSOLUTE POSITION CONVERSION

# 3 TRACK DATABASE MANAGER

## 3.1 TRACK DATABASE UPDATE

3.1.1 DATABASE MESSAGE CONTROL
3.1.2 DATABASE FILTER
3.1.3 PRIORITIZE TUPLES
3.1.4 WRITE TUPLE TO DATABASE
3.1.5 REMOVE TRACK TUPLE
3.1.6 CHANGE ATTRIBUTE VALUE

## 3.2 TRACK REQUEST

3.2.1 MANAGE TRACK DATABASE REQUEST
3.2.2 ACCESS TRACK TUPLE
3.2.3 FORWARD TRACK TUPLE

## 3.3 DATABASE MONITOR

3.3.1 SCAN TRACK DATABASE
3.3.2 TIMEOUT
3.3.3 ARCHIVE TRACKS
3.3.4 CONSTRAINT VIOLATED
3.3.5 IDENTIFY SIMILARITIES
3.3.6 MODIFY TRACK DATABASE
3.3.7 MONITOR SETUP

## 3.4 OWNSHIP TRACK MONITOR

3.4.1 OWNSHIP NAVIGATION MONITOR
3.4.2 OWNSHIP TRACK GENERATOR

# 4 TRACK CONTROLLER

## 4.1 TRACK MANAGER DIALOGUE

### 4.1.1 INITIALIZE CONSTRAINTS

4.1.1.1 CONSTRAINT SELECTION
4.1.1.2 TRANSMISSION SEQUENCE MENU
4.1.1.3 ARCHIVE SETUP MENU
4.1.1.4 TRACK MONITOR MENU
4.1.1.5 TRACK FILTER MENU

### 4.1.2 DATABASE MANIPULATION

4.1.2.1 DATABASE FUNCTION SELECTION
4.1.2.2 TRACK DISPLAY MENU
4.1.2.3 ADD TRACK MENU
4.1.2.4 UPDATE TRACK MENU
4.1.2.5 DELETE TRACK MENU

## 4.2 RETRIEVE TRACK TUPLE INFORMATION

## 4.3 ADD NEW TRACK TO DATABASE

## 4.4 MODIFY EXISTING TRACK DATA

## 4.5 DELETE TRACK FROM DATABASE

## 4.6 TRACK MANAGER DISPLAY

4.6.1 DISPLAY RESOLUTION SCREEN
4.6.2 DISPLAY INTEL REPORT SCREEN
4.6.3 DISPLAY TRACK TUPLE SCREEN

# 5 TACTICAL COMMAND DISPLAYS

## 5.1 BATTLE MANAGER DIALOGUE

### 5.1.1 SYSTEM INPUT

#### 5.1.1.1 FUNCTION SELECTION

5.1.1.1.1 BATTLE MANAGER FUNCTION SELECTION
5.1.1.1.2 STATUS MENU
5.1.1.1.3 TRACK PLOT MENU
5.1.1.1.4 GENERATE MESSAGE MENU
5.1.1.1.5 VIEW MESSAGE MENU

#### 5.1.1.2 NETWORK CONSTRAINT SELECTION

5.1.1.2.1 NETWORK COMMAND OPTIONS
5.1.1.2.2 STATUS REPORT MENU
5.1.1.2.3 NETWORK SETUP MENU
5.1.1.2.4 EMISSIONS STATUS MENU

### 5.1.2 SYSTEM OUTPUT

5.1.2.1 DISPLAY EMERGENCY STATUS REPORT
5.1.2.2 MESSAGE ARRIVAL DISPLAY
5.1.2.3 DISPLAY WEAPON STATUS
5.1.2.4 TRACK DISPLAY
5.1.2.5 DISPLAY EDIT SCREEN
5.1.2.6 DISPLAY TEXT FILE

## 5.2 PLATFORM STATUS MONITOR

## 5.3 GRAPHICAL TRACK DISPLAY

5.3.1 TRACK DISPLAY MONITOR
5.3.2 MAP GENERATOR
5.3.3 WINDOW GENERATOR
5.3.4 TRACK DISPLAY GENERATOR
5.3.5 GEOMETRIC DISPLAY GENERATOR

## 5.4 MESSAGE GENERATOR

### 5.4.1 EDIT DIALOGUE

5.4.1.1 MESSAGE SELECTION
5.4.1.2 RETRIEVE TEMPLATE
5.4.1.3 RETRIEVE EXISTING FILE
5.4.1.4 TEXT EDITOR
5.4.1.5 RETRIEVE TRACK TEXT DATA
5.4.1.6 SEND TEXT FILE
5.4.1.7 SAVE TEXT FILE

5.4.2 CREATE NEW FILE

5.4.3 READ EXISTING FILE

5.5 MESSAGE PROCESSOR

5.6 TEXTUAL MESSAGE DISPLAY

5.6.1 MESSAGE RETRIEVAL
5.6.2 INCOMING MESSAGE QUEUE

5.7 STATUS REPORT GENERATOR

# 6 WEAPONS SYSTEMS INTERFACE

## 6.1 OWNSHIP WEAPONS STATUS MONITOR

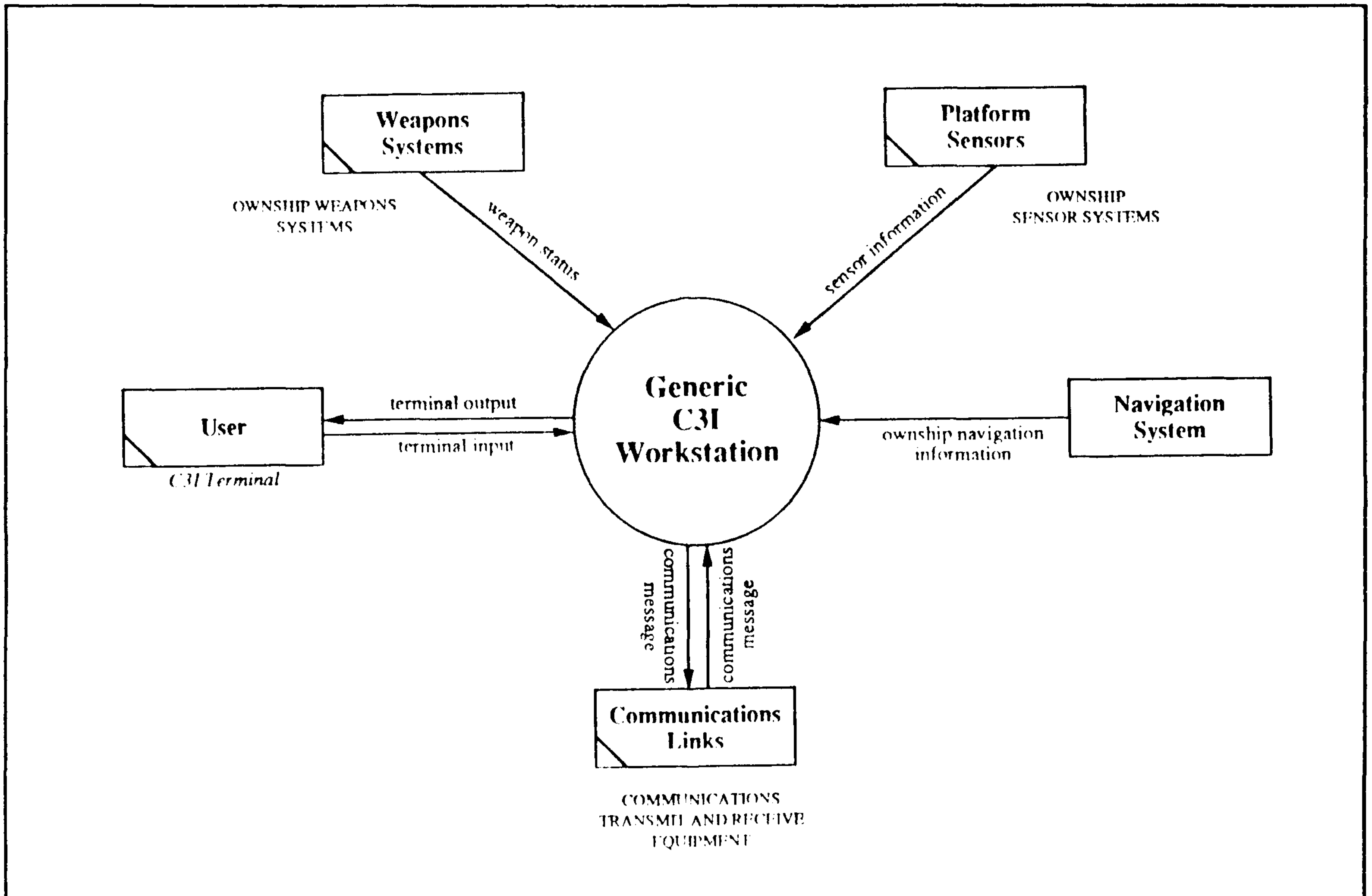
## 6.2 EMERGENCY STATUS REPORTER

## 6.3 CIWS STATUS CONTROL

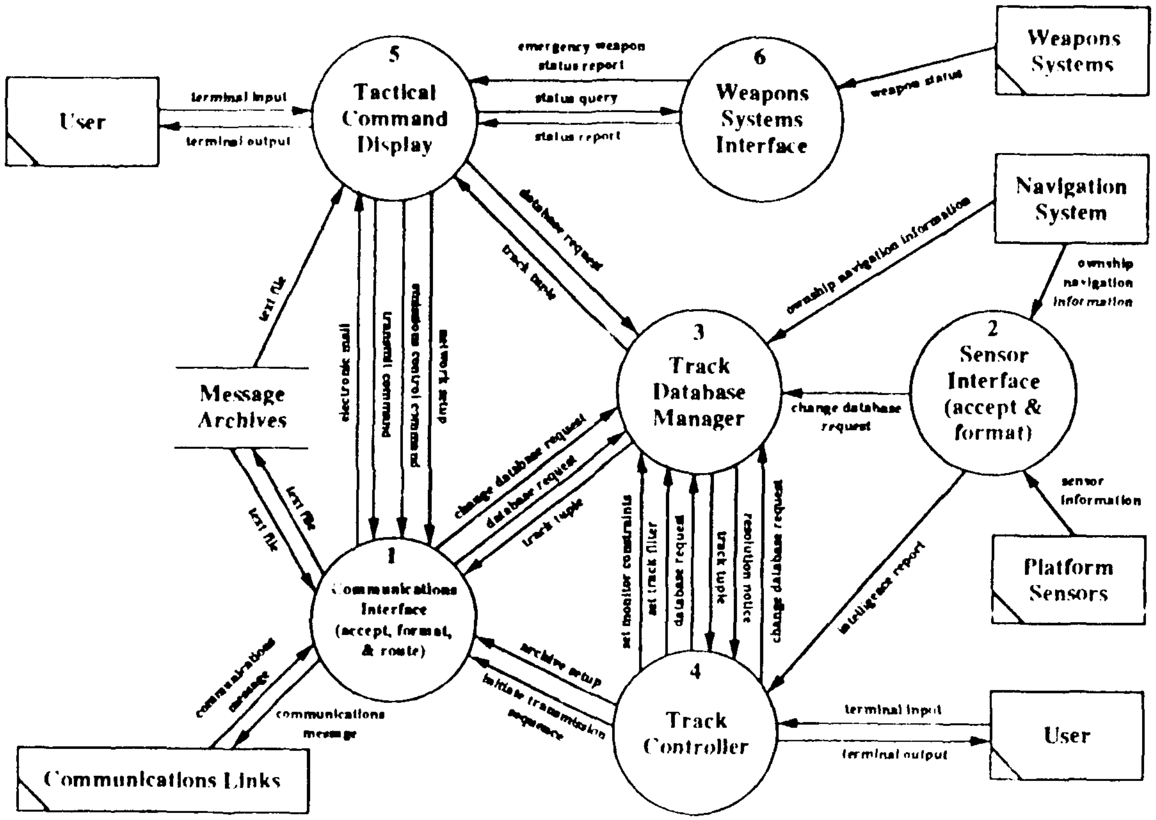
## 6.4 5"/54 GUN WEAPON SYSTEM STATUS CONTROL

## 6.5 TOMAHAWK WEAPON STATUS CONTROL

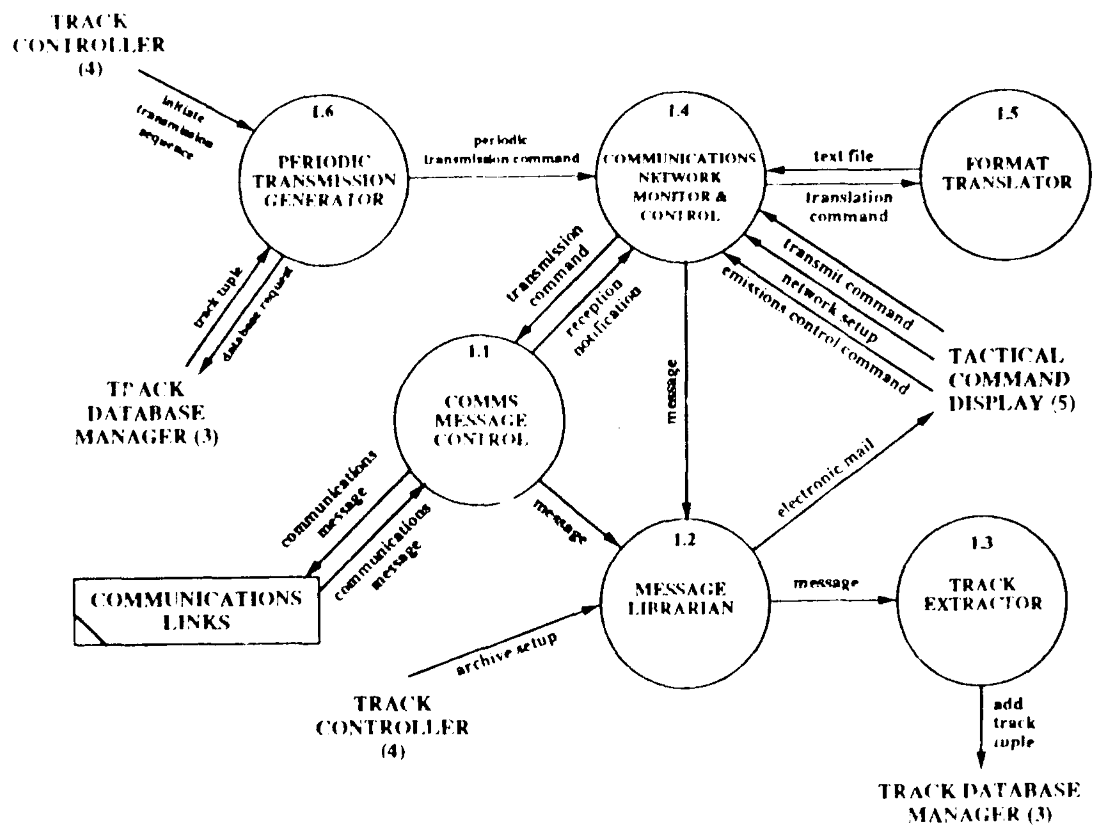
## 6.6 MK46 TORPEDO STATUS CONTROL




# Context Diagram

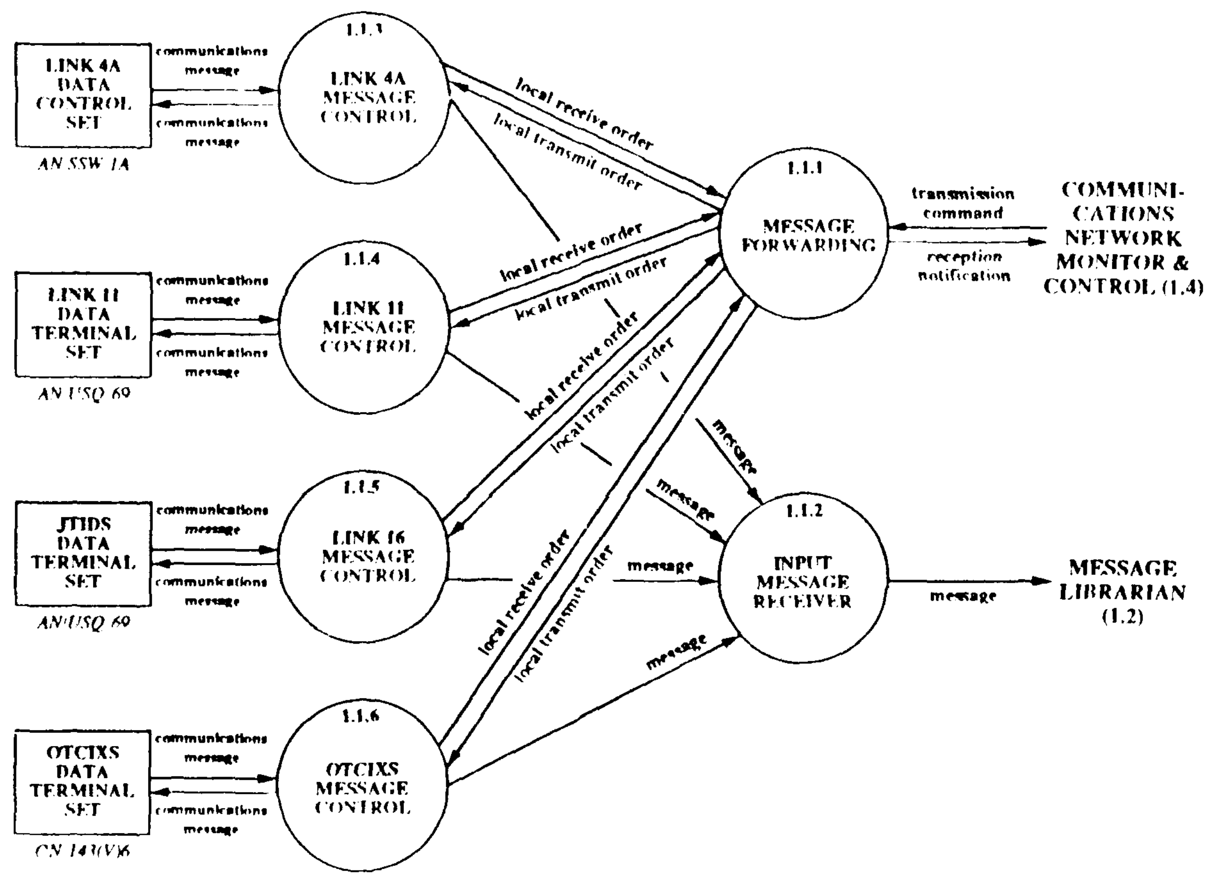



# Expanded Context Diagram



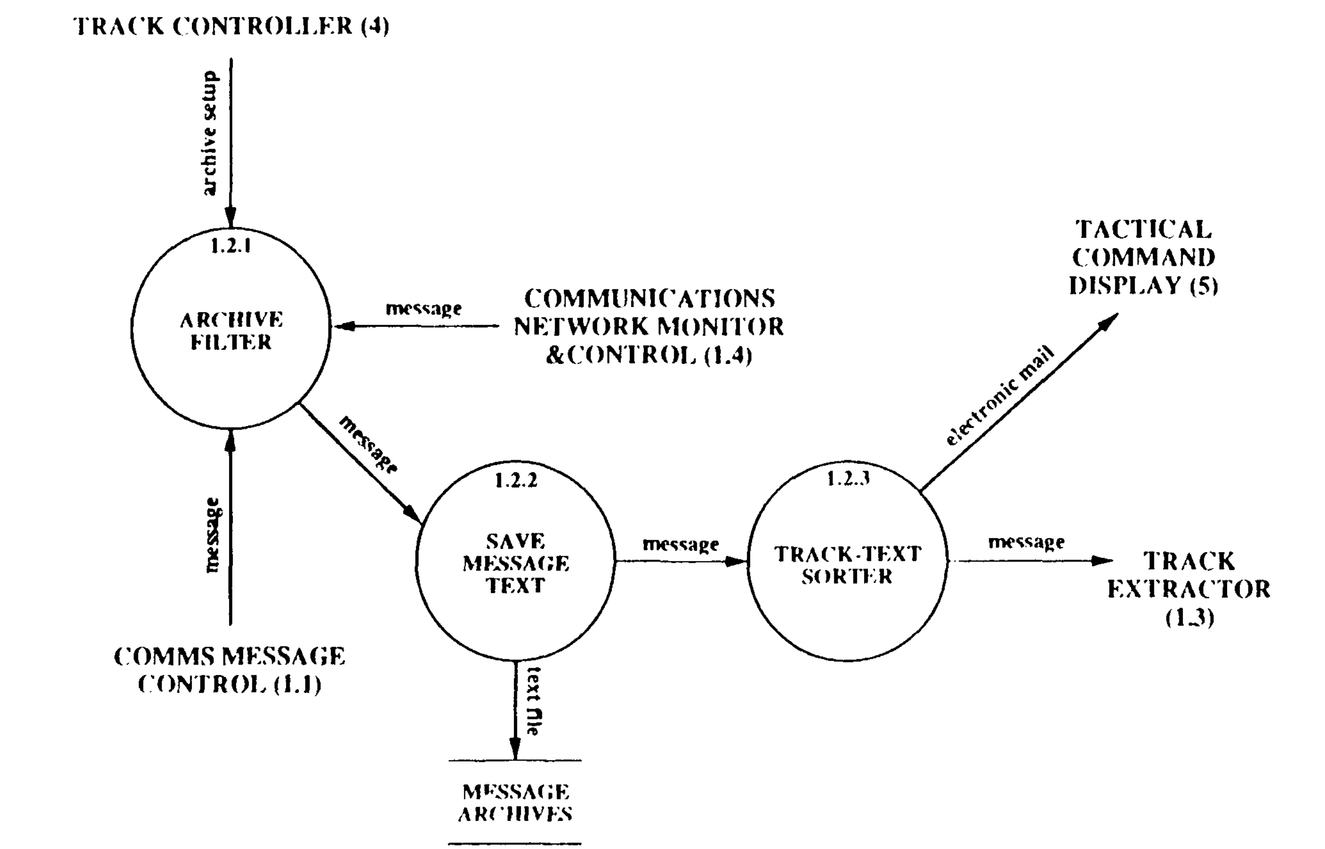

Communications Interface (accept, format & route)

Expansion of Bubble 1

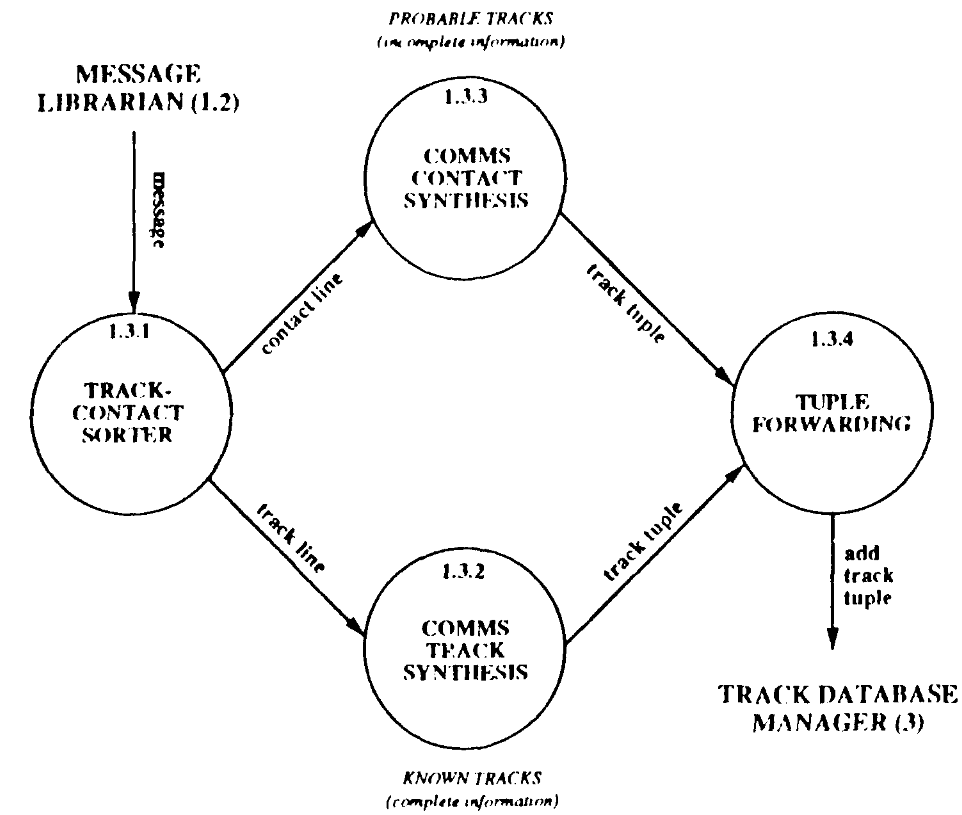



Communications Message Control

Expansion of Bubble 1.1

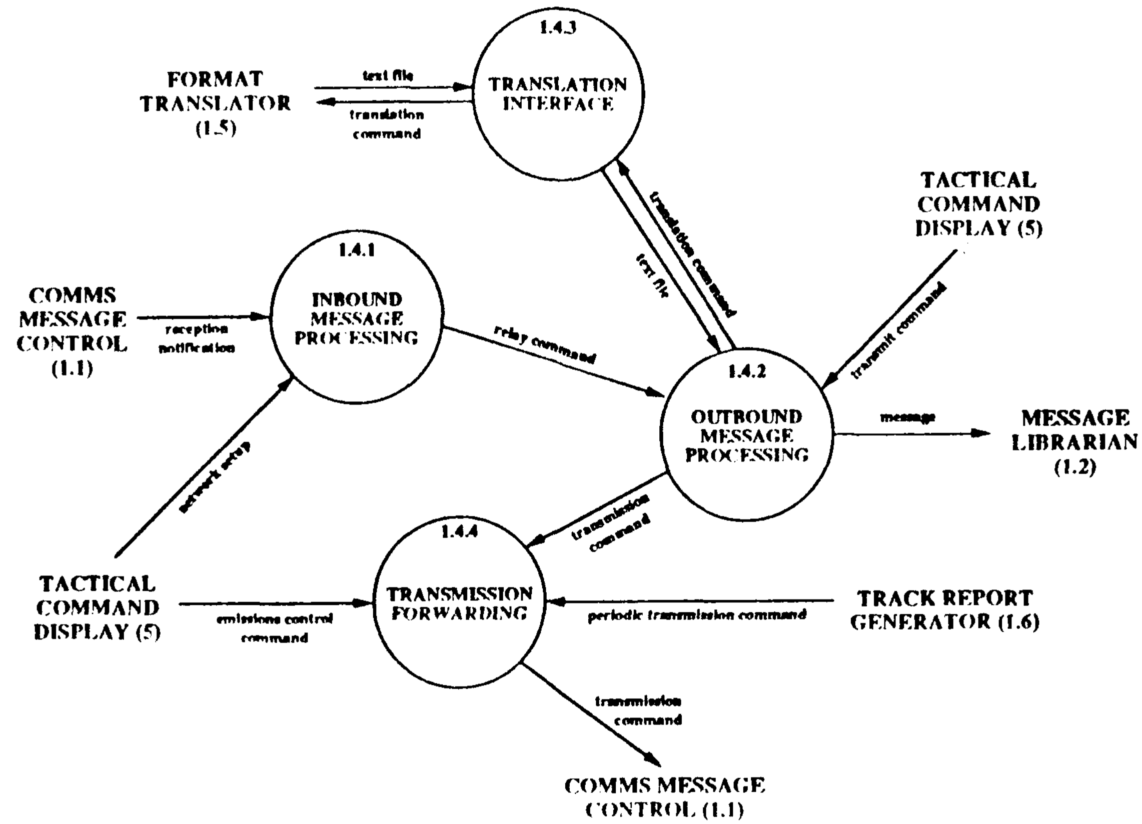



| | |
|---|---|
| **Message Librarian** | Expansion of Bubble **1.2** |

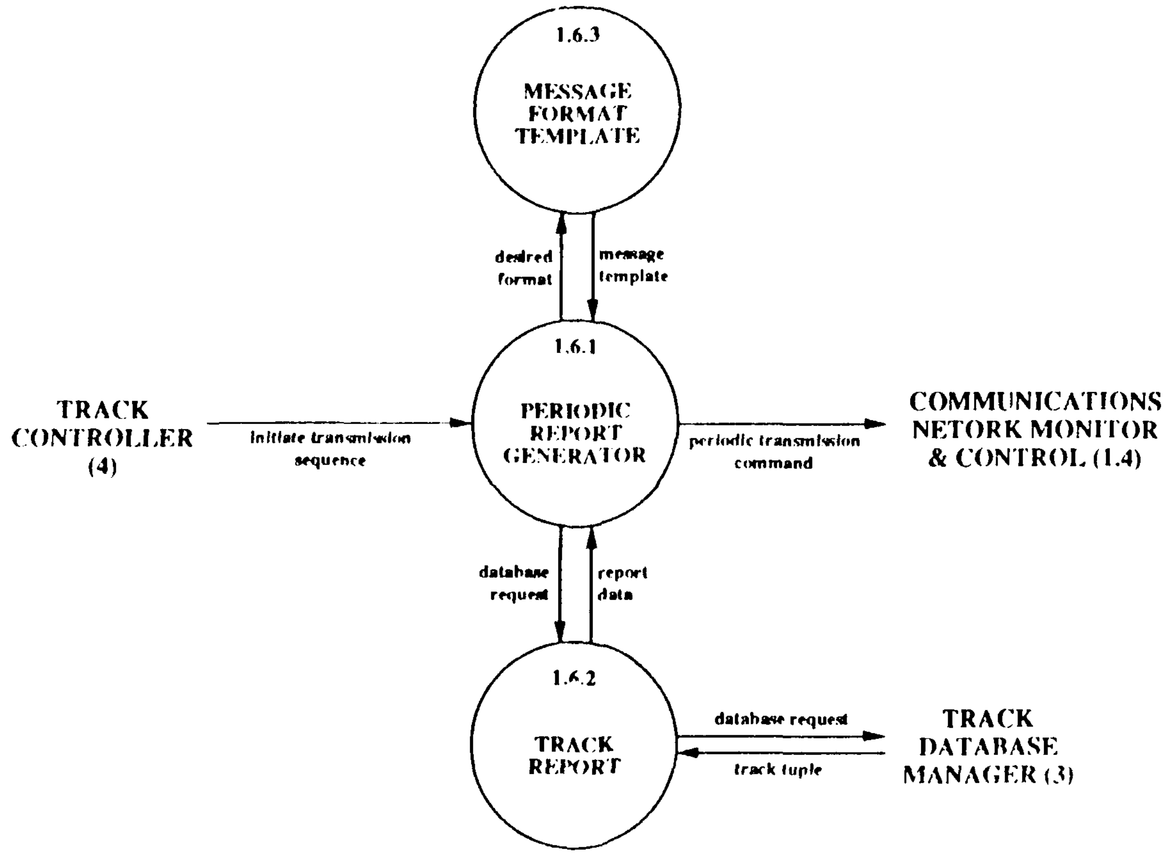



**Track Extractor**

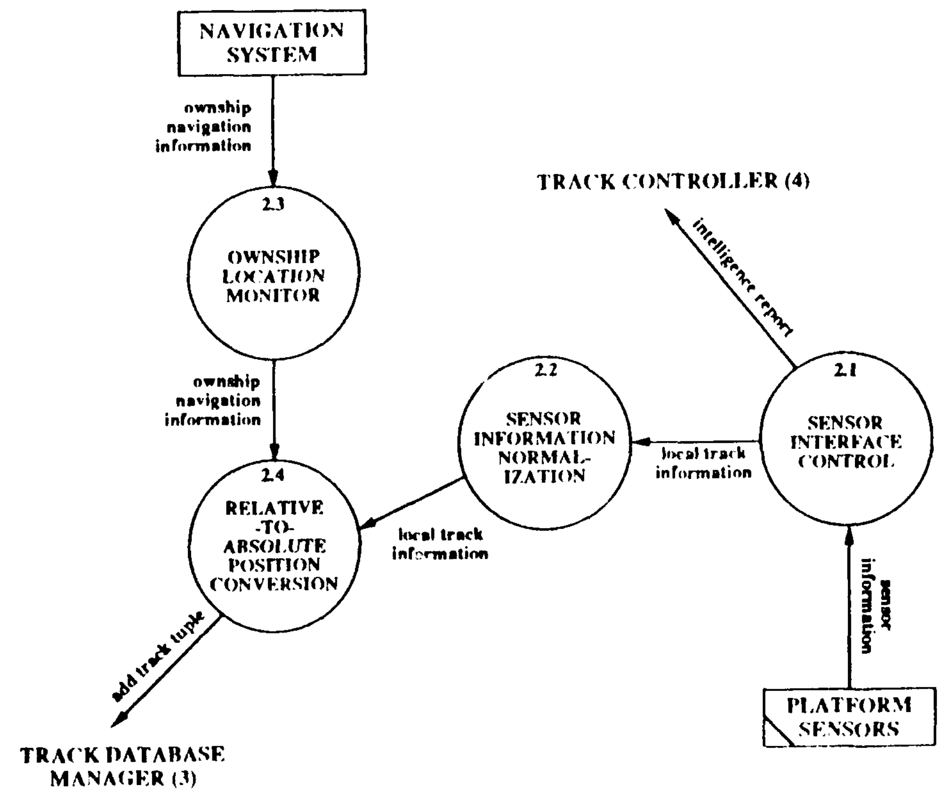
Expansion of Bubble **1.3**




| Communications Network Monitor & Control | Expansion of Bubble | 1.4 |
|---|---|---|

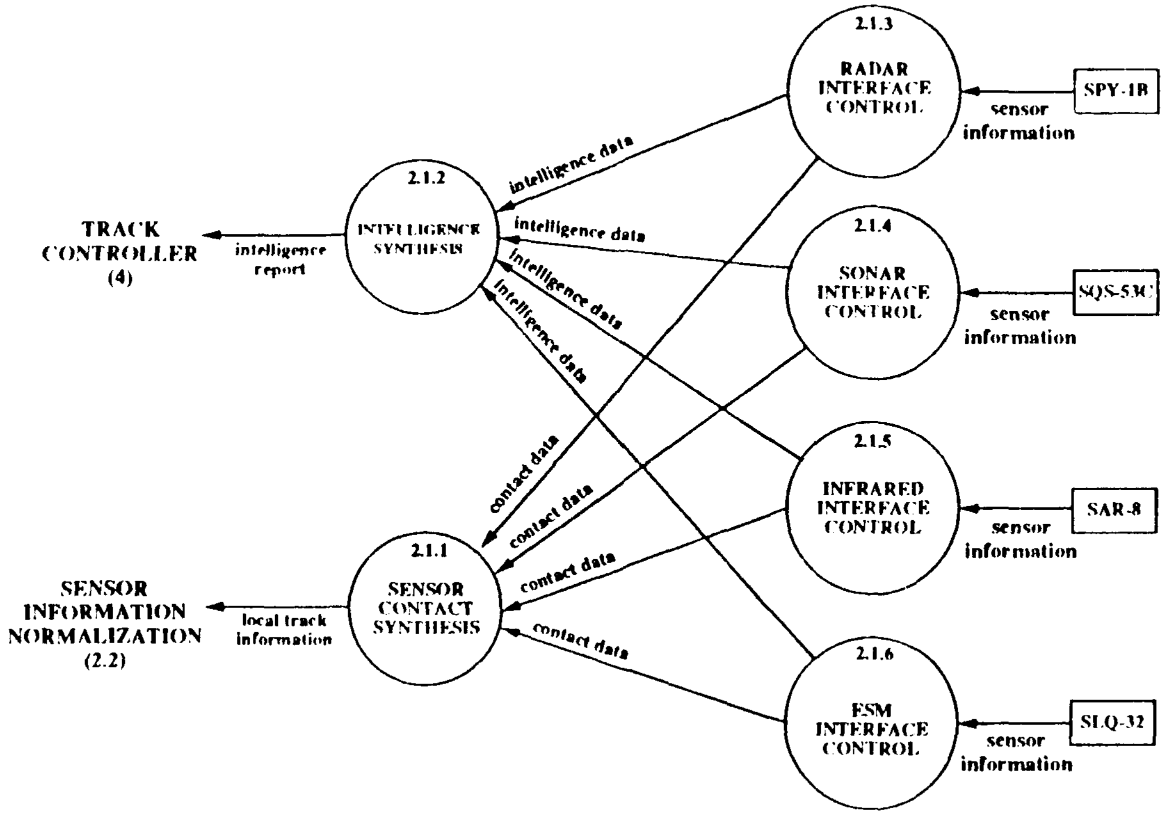



**Periodic Transmission Generator** | Expansion of Bubble **1.6**



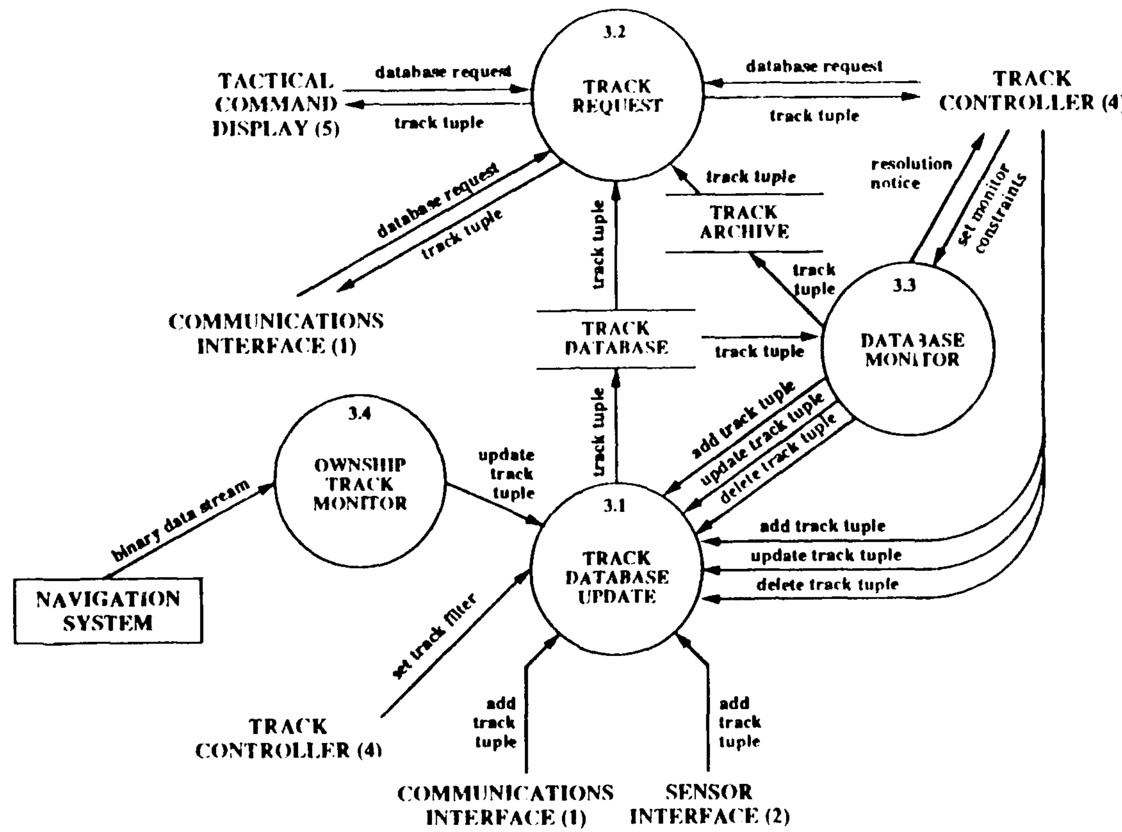

**Sensor Interface (accept & format)**

Expansion of Bubble 2

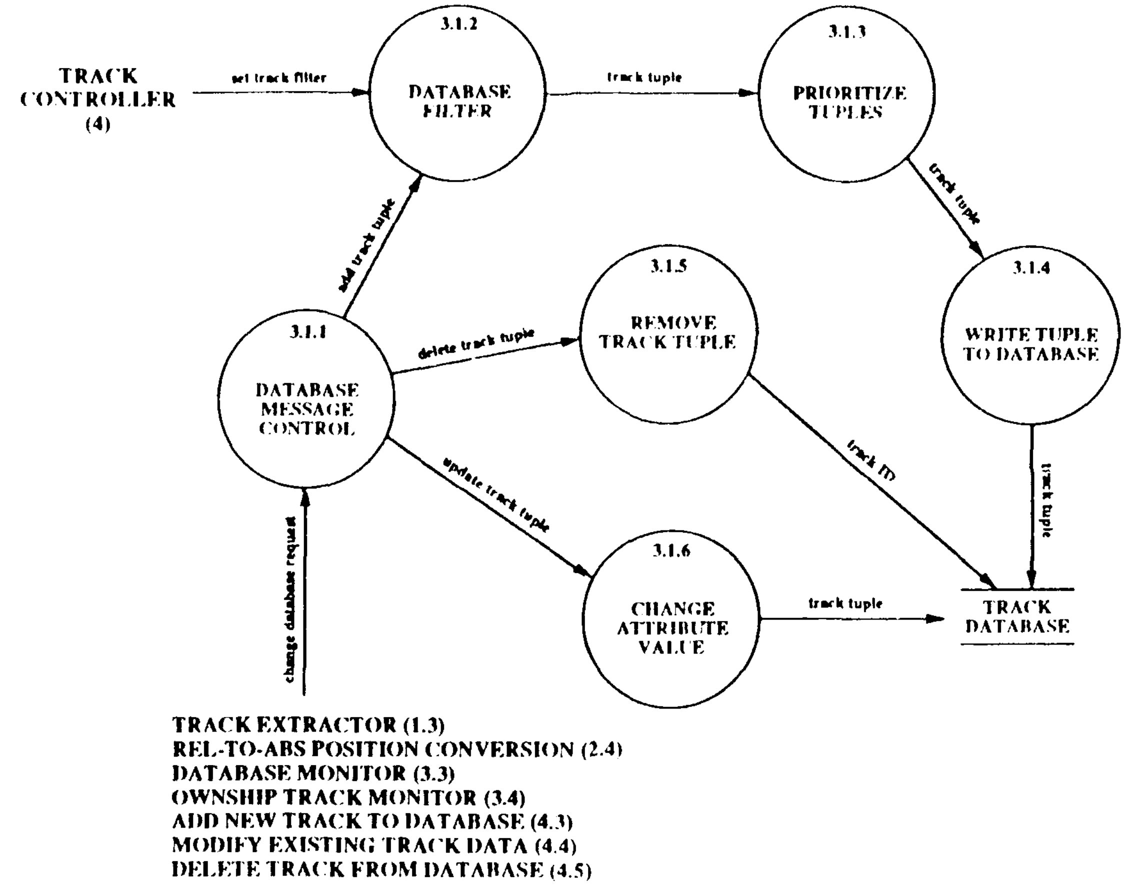



Sensor Interface Control

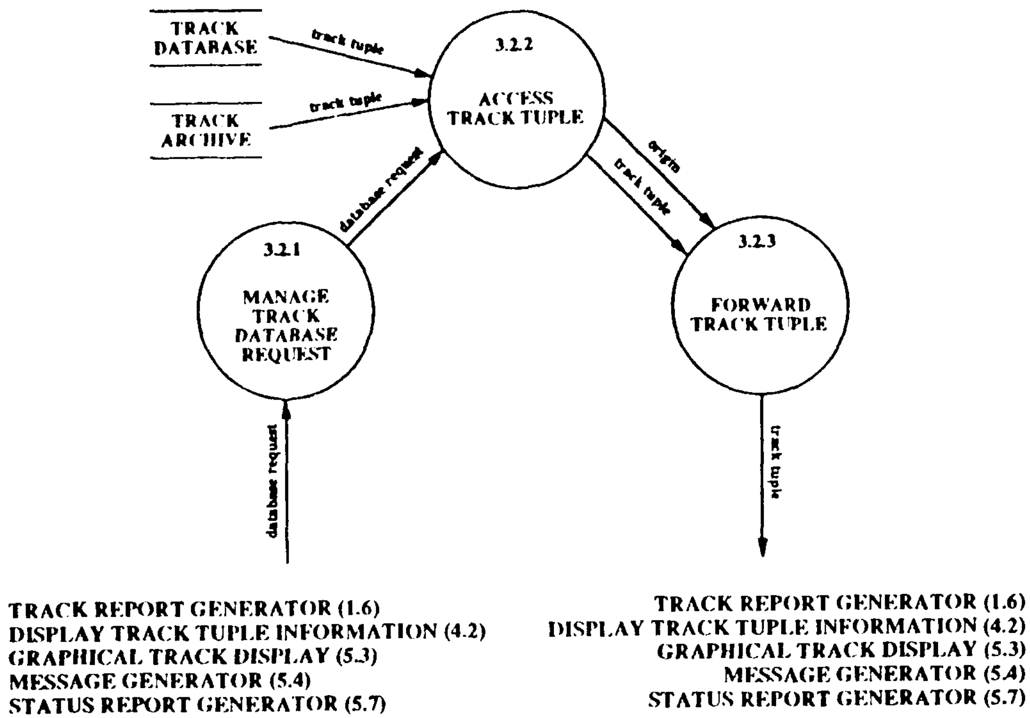
Expansion of Bubble 2.1



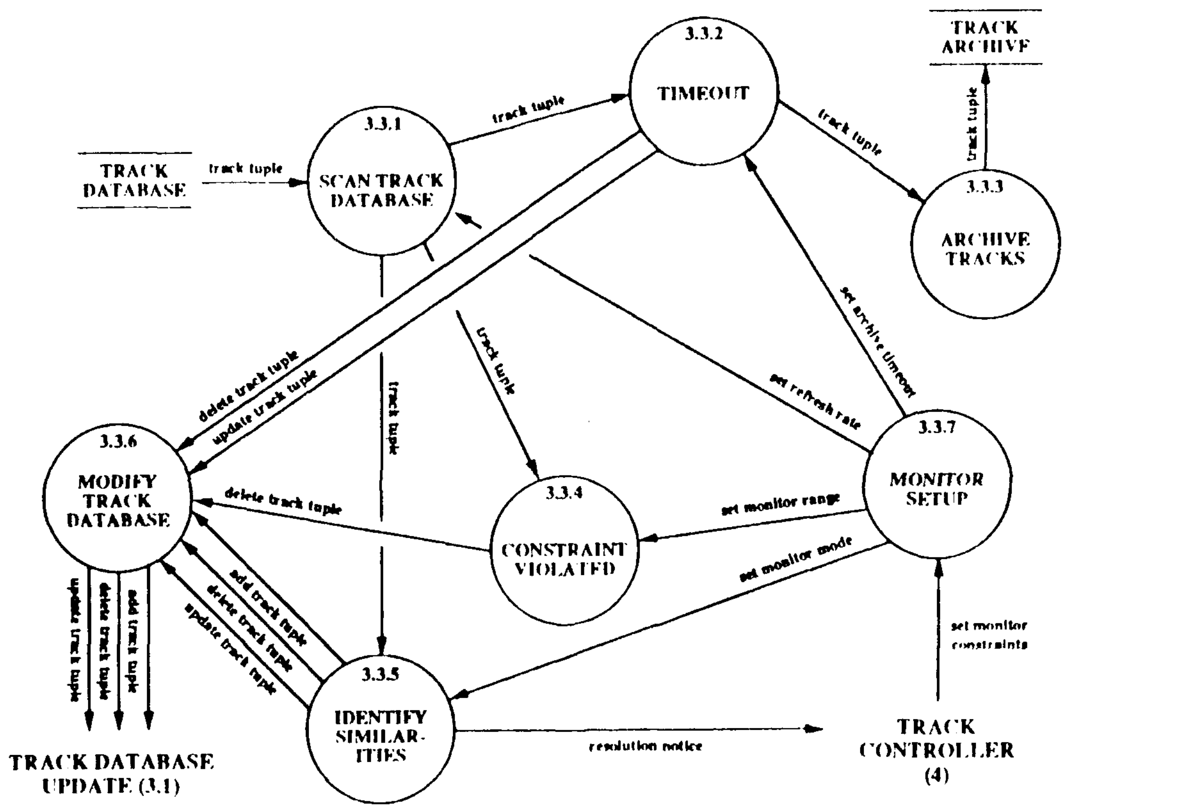

**Track Database Manager** | Expansion of Bubble 3




**Track Database Update** | Expansion of Bubble **3.1**

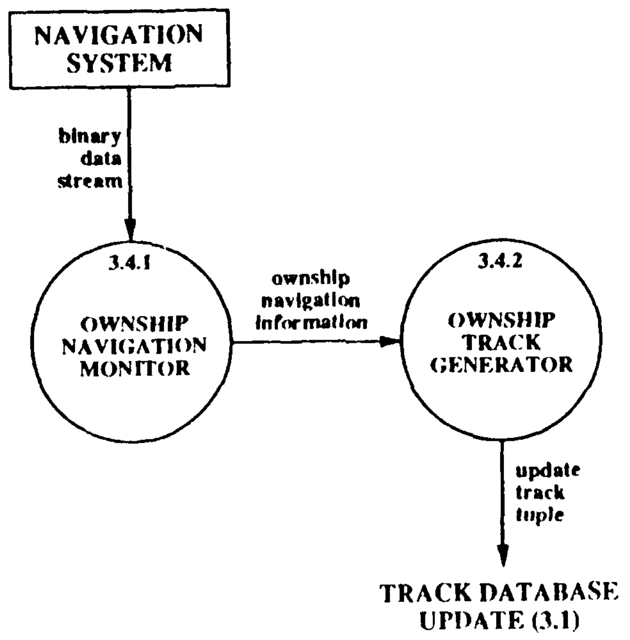



| | |
|---|---|
| **Track Request** | Expansion of Bubble **3.2** |

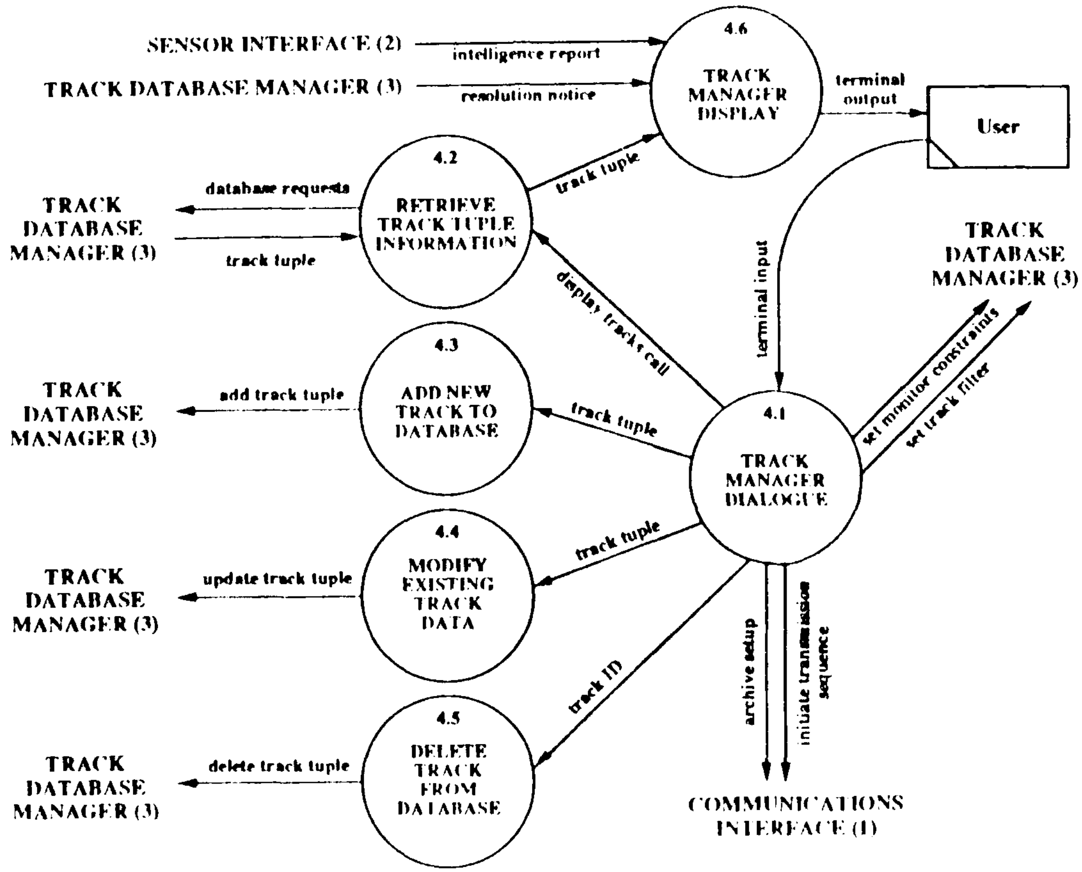



**Database Monitor**

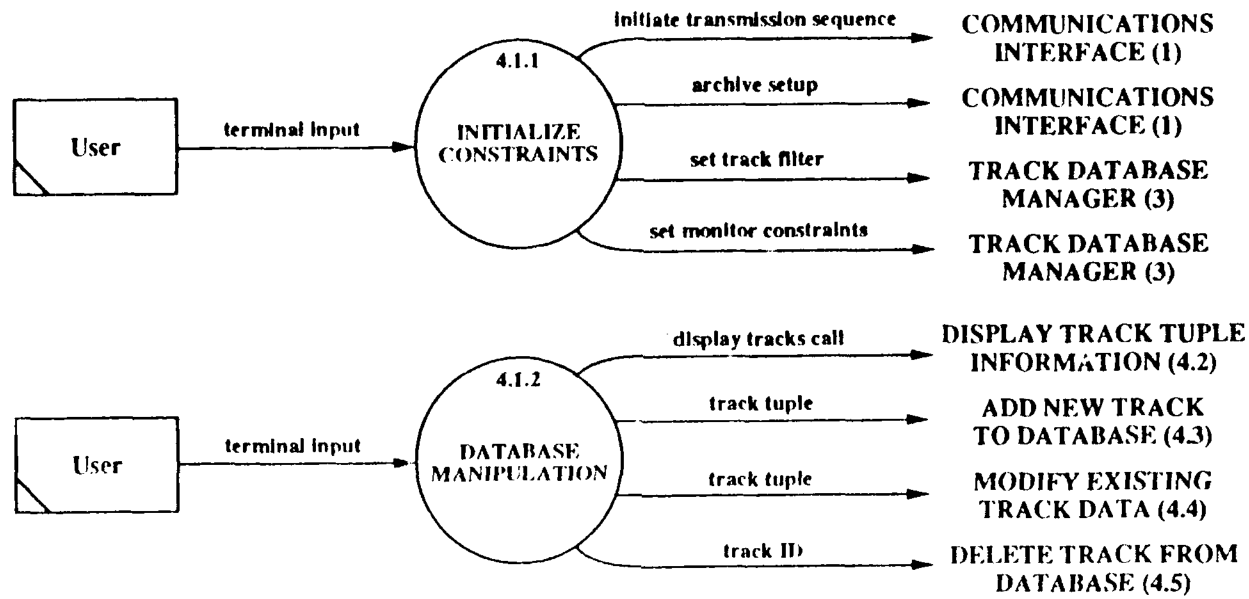
Expansion of Bubble **3.3**



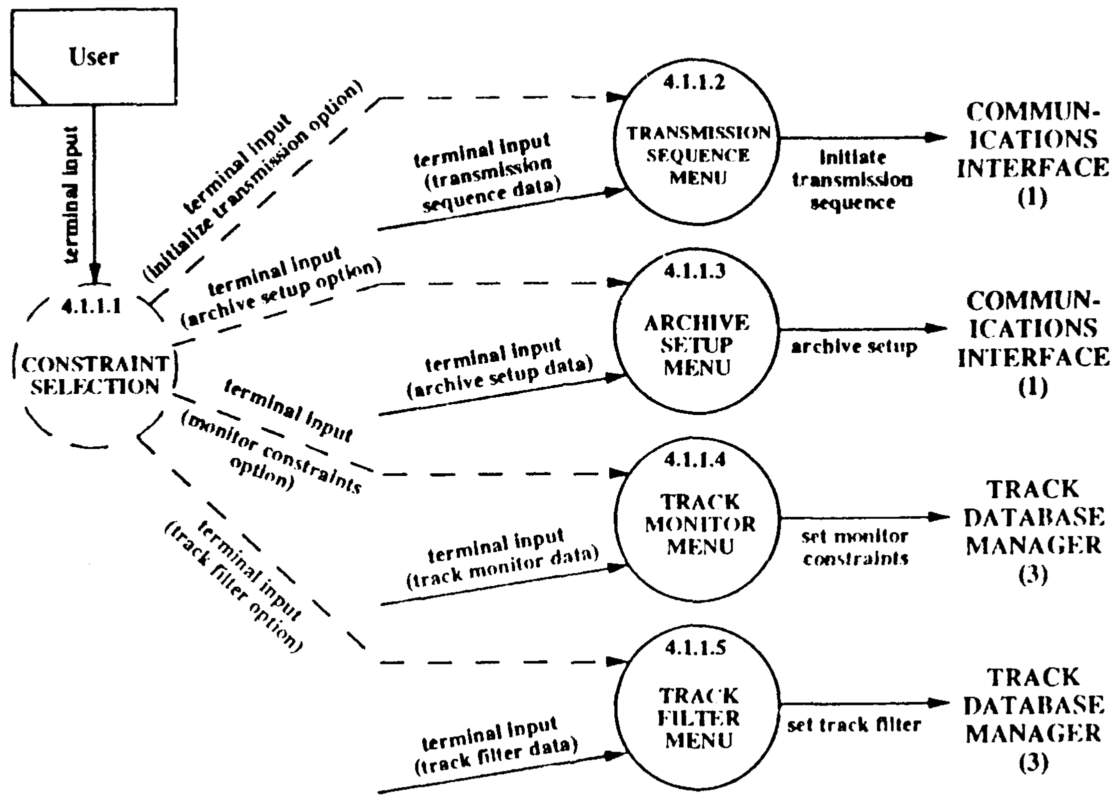

Ownship Track Monitor | Expansion of Bubble 3.4




**Track Controller** | Expansion of Bubble **4**

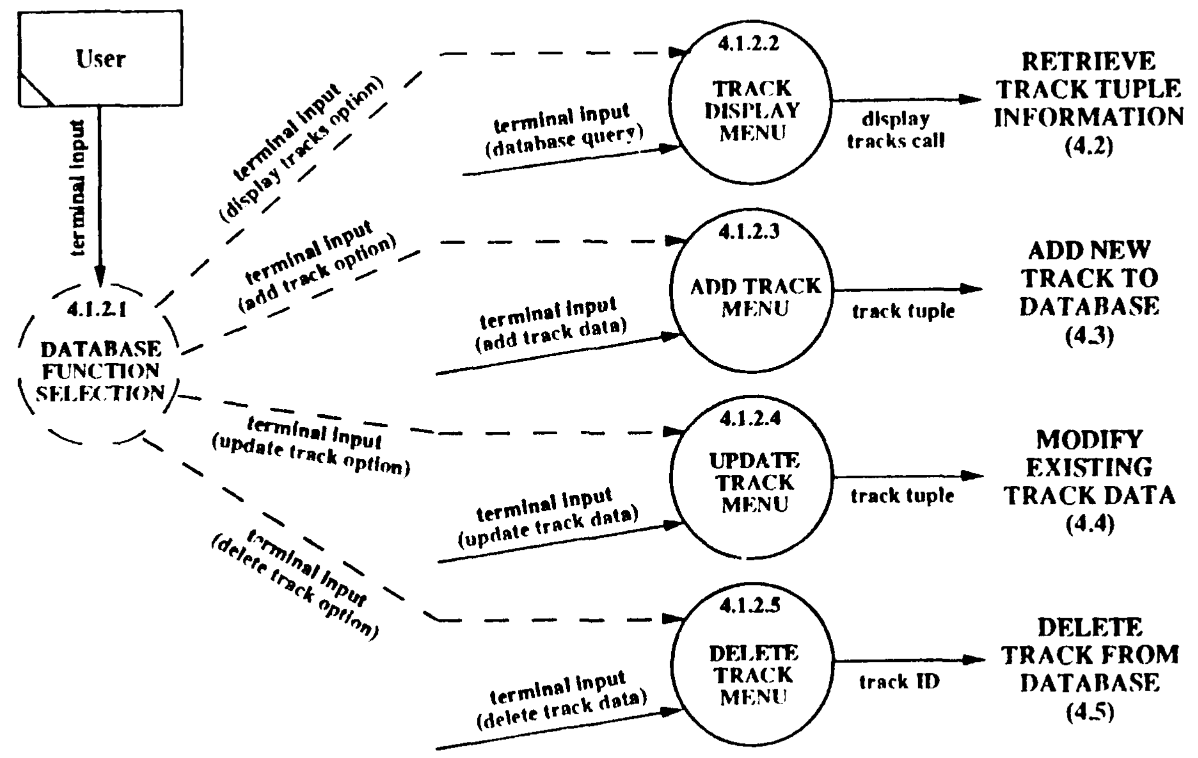



| | |
|---|---|
| **Track Manager Dialogue** | Expansion of Bubble **4.1** |



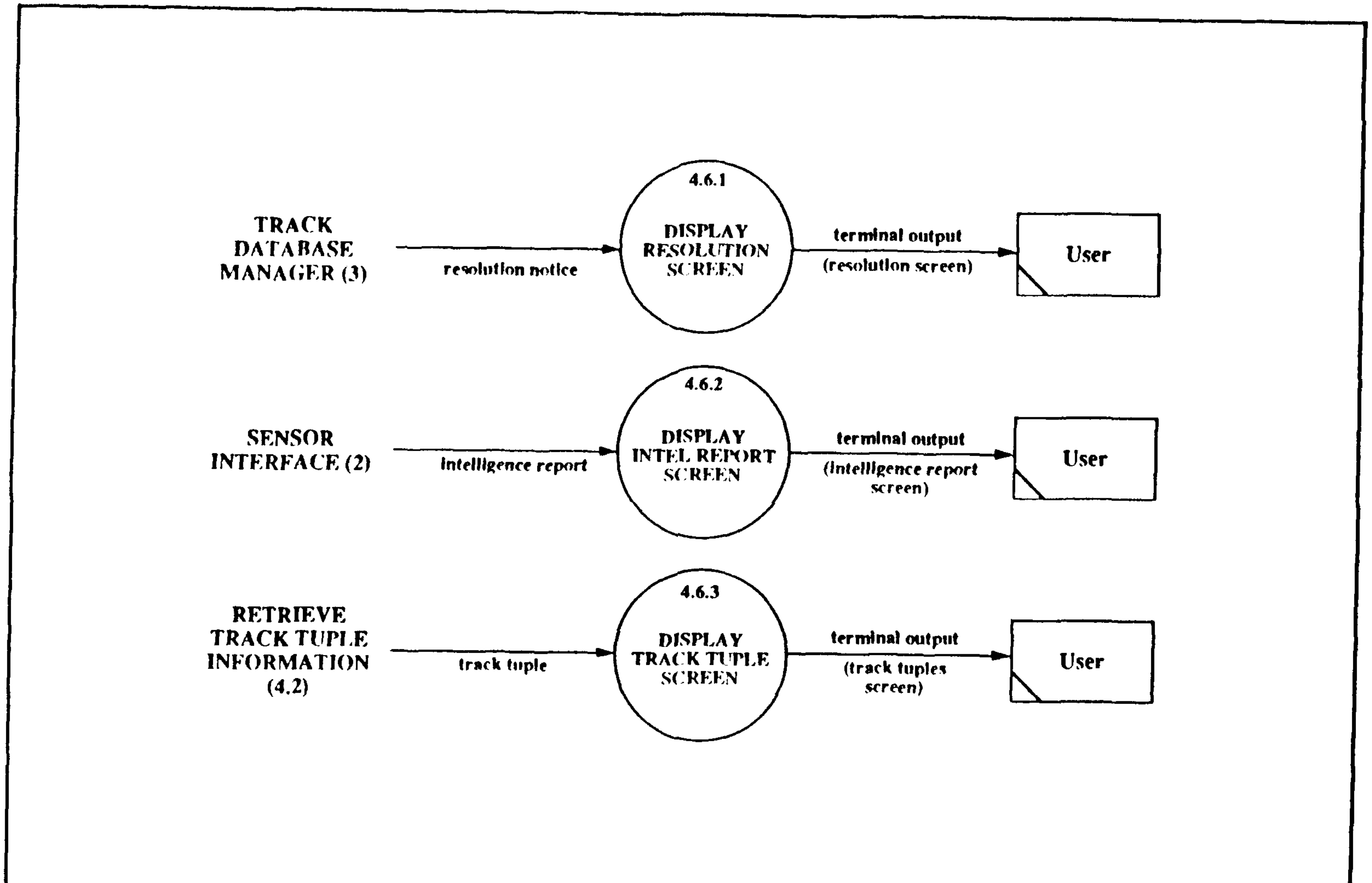

**Initialize Constraints**

Expansion of Bubble **4.1.1**

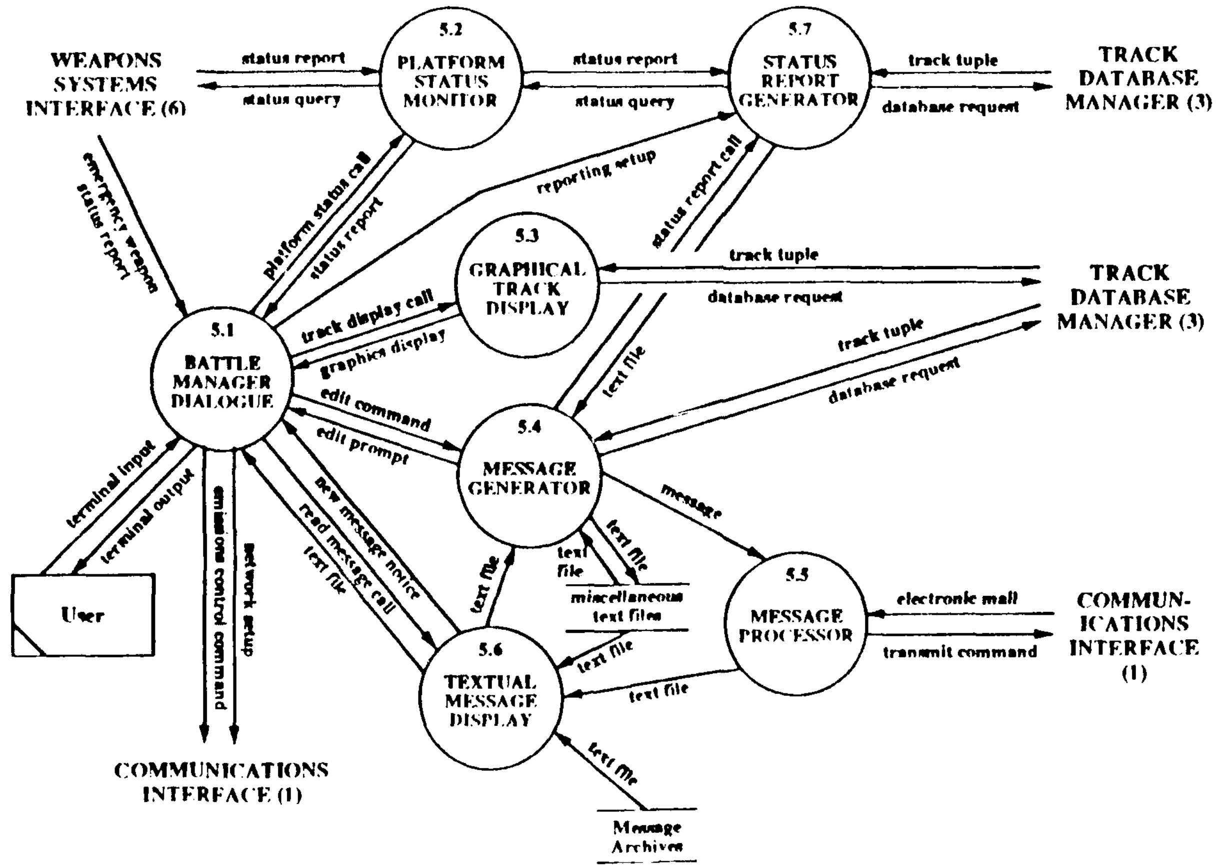



Database Manipulation

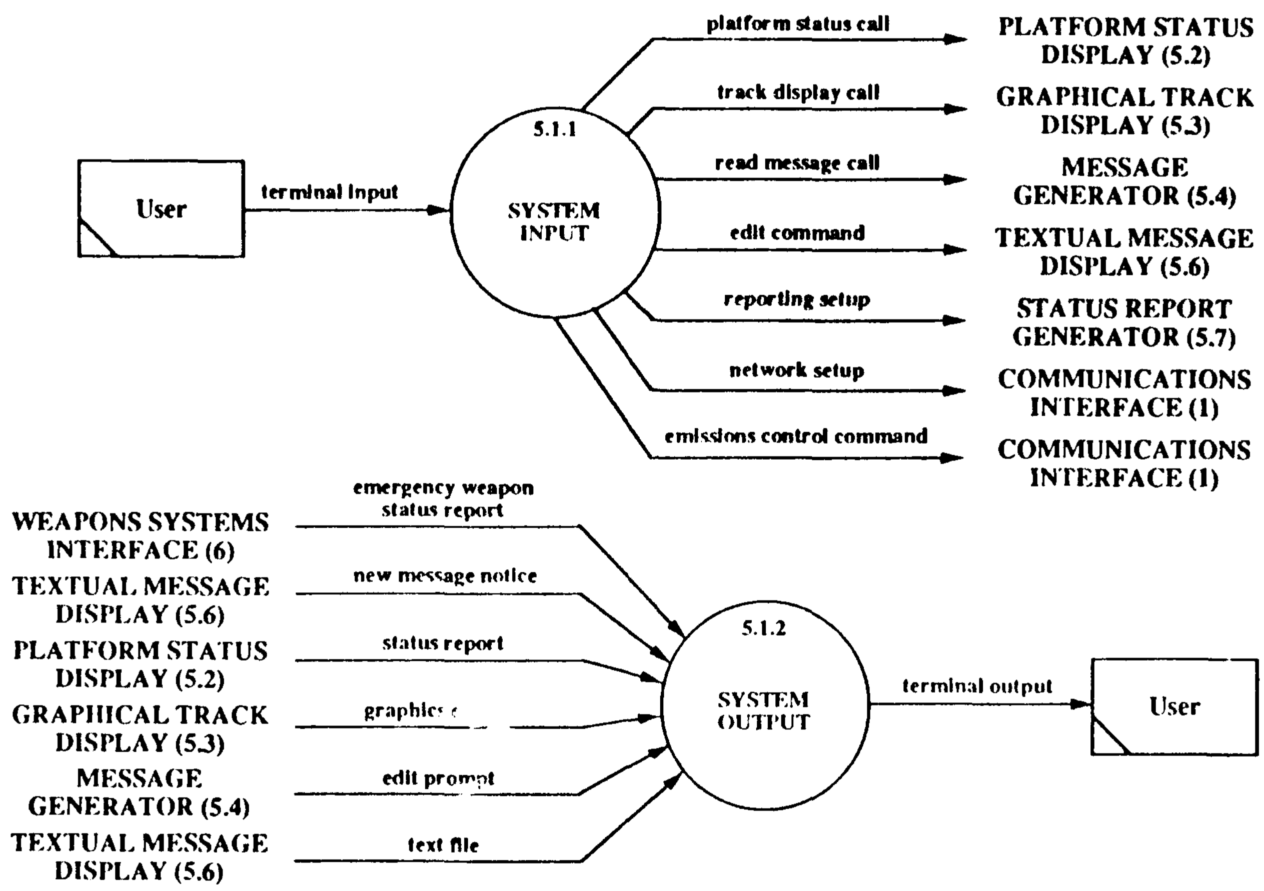
Expansion of Bubble 4.1.2



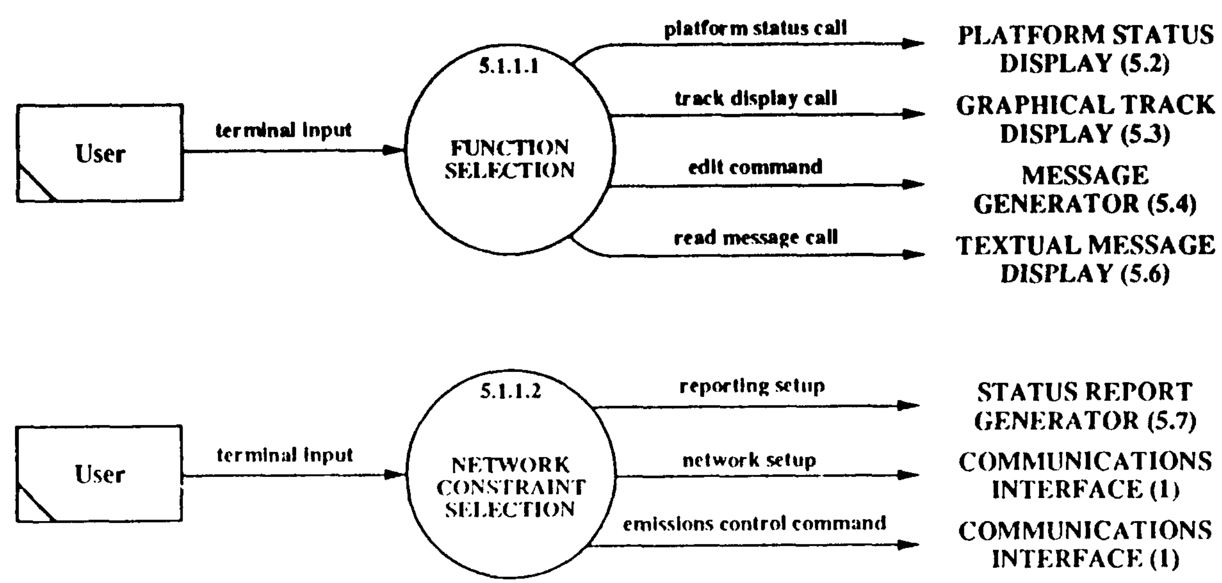

**Track Manager Display** | Expansion of Bubble **4.6**




| Tactical Command Display | Expansion of Bubble 5 |
|---|---|

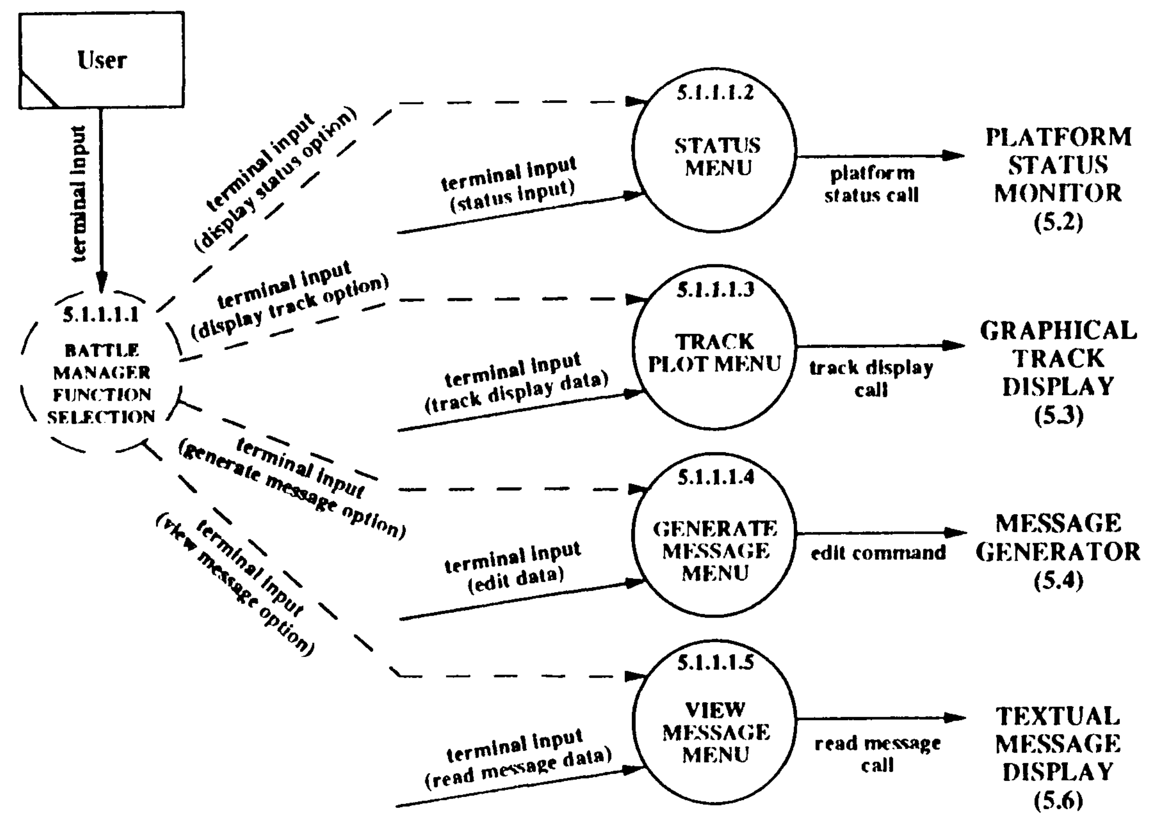



| | |
|---|---|
| **Battle Manager Dialogue** | Expansion of Bubble **5.1** |

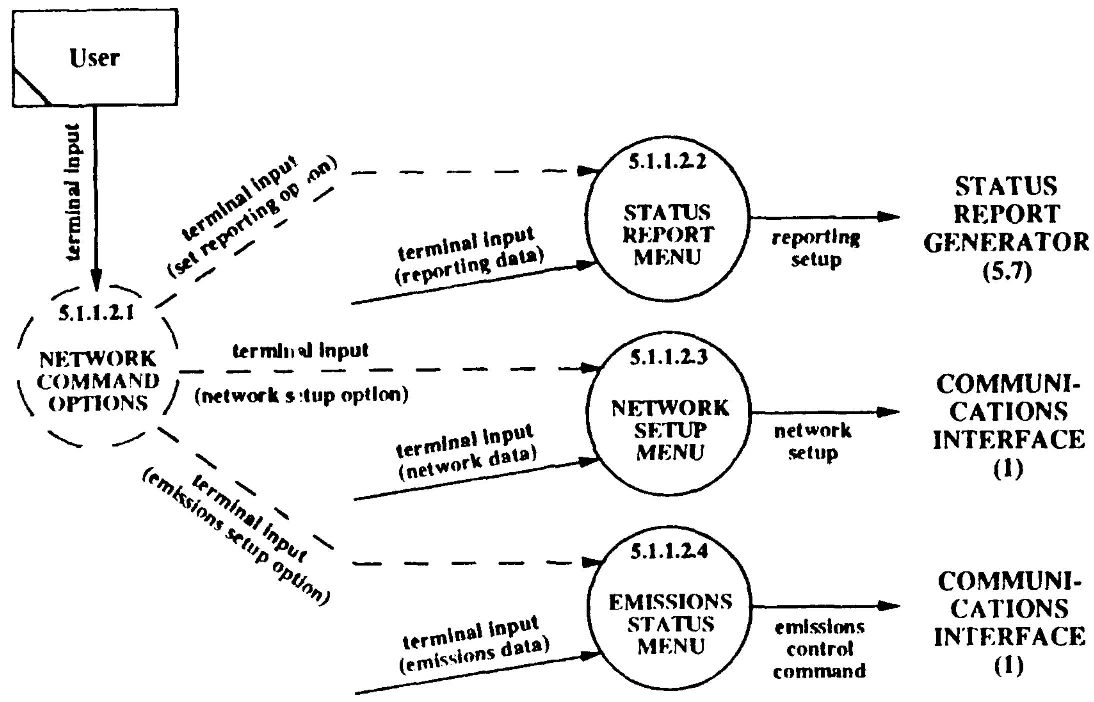



System Input

Expansion of Bubble **5.1.1**

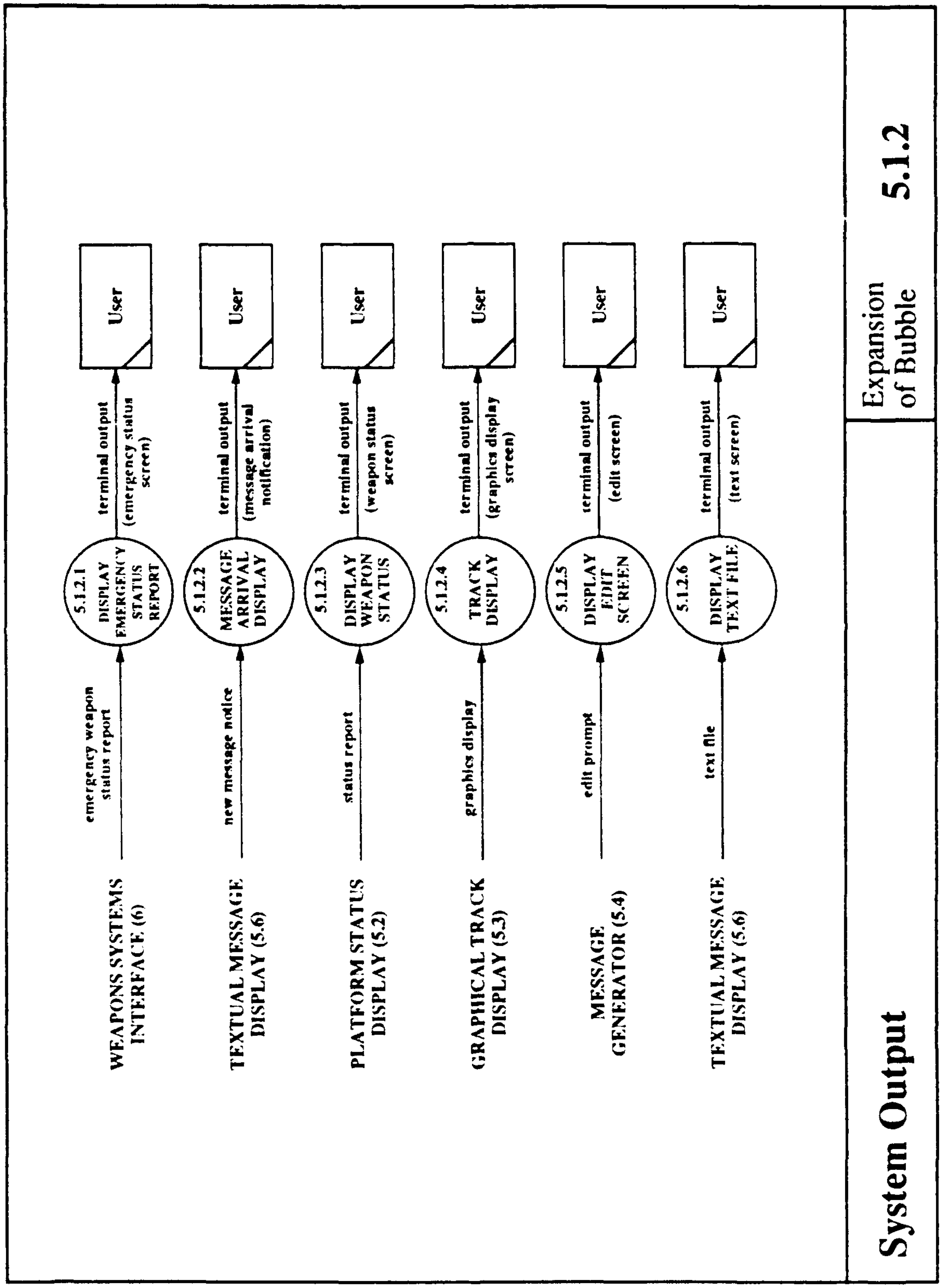



| **Function Selection** | Expansion of Bubble **5.1.1.1** |
|---|---|

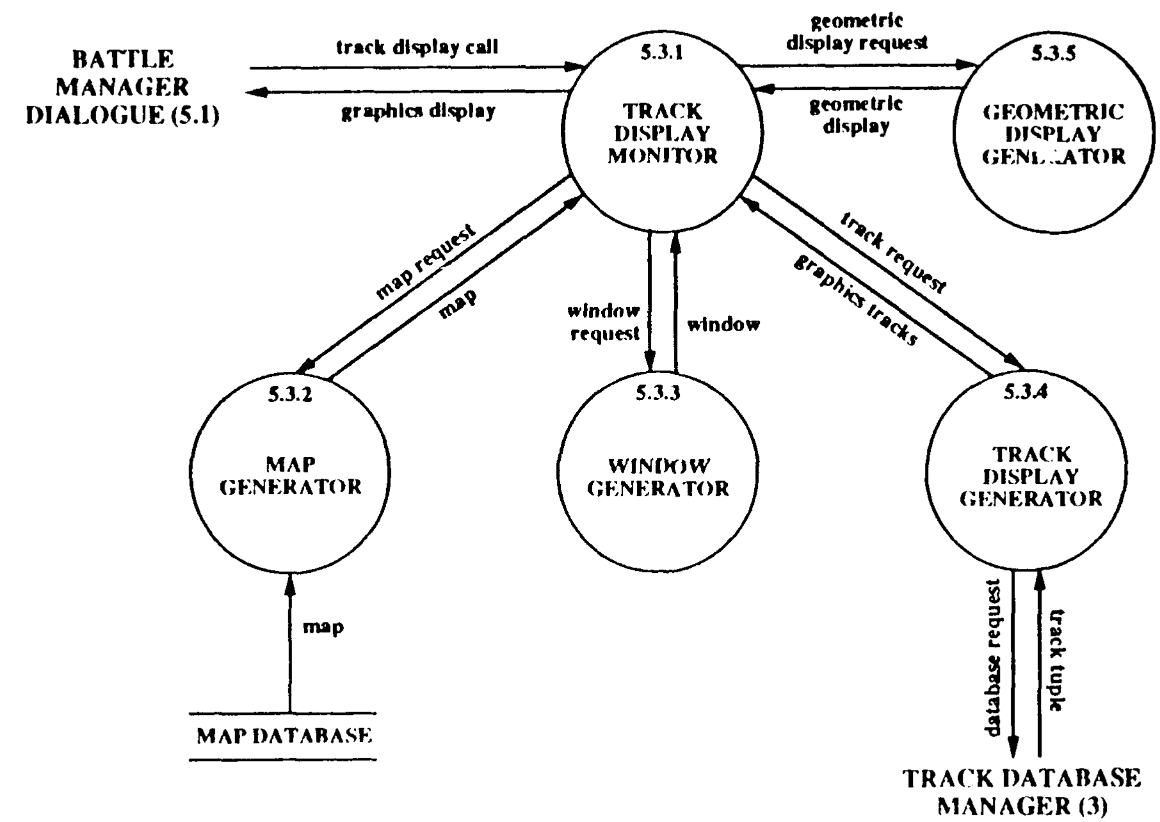



**Network Constraint Selection**

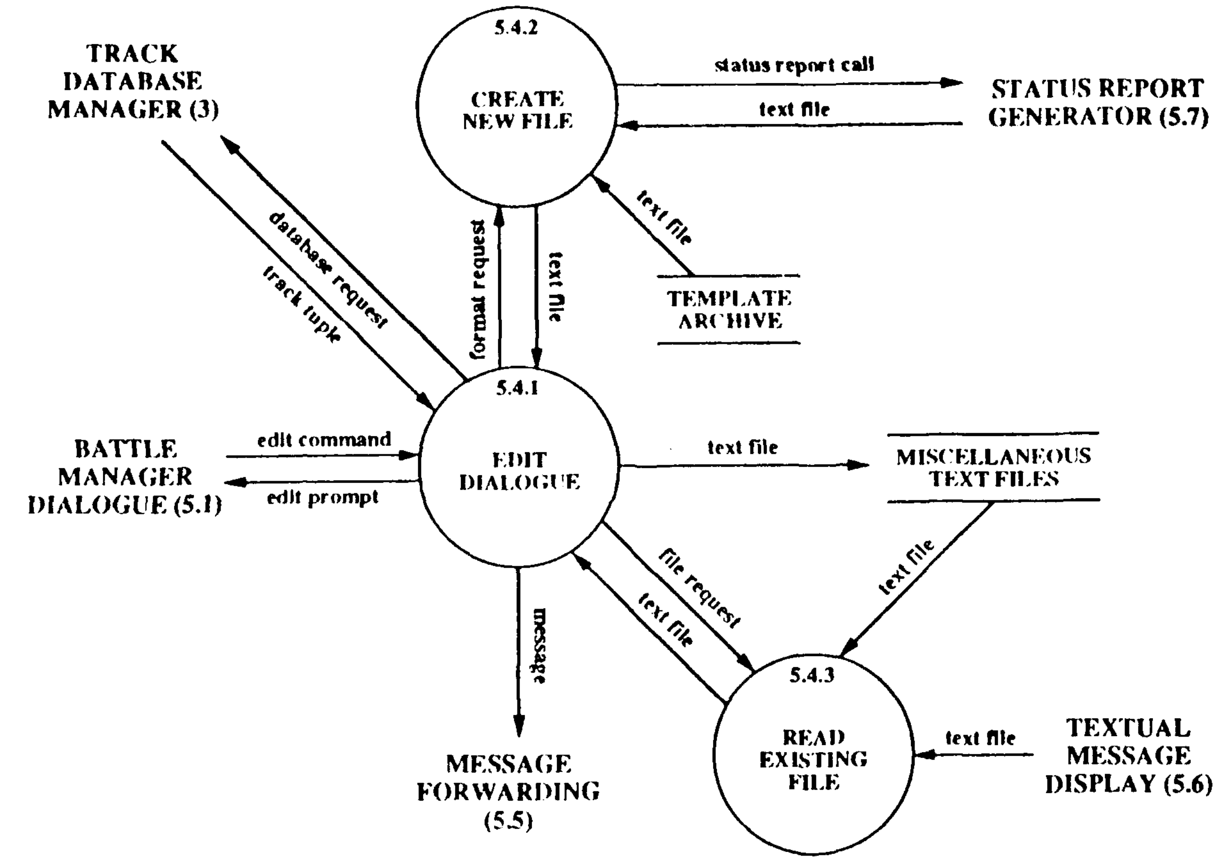
Expansion of Bubble **5.1.1.2**



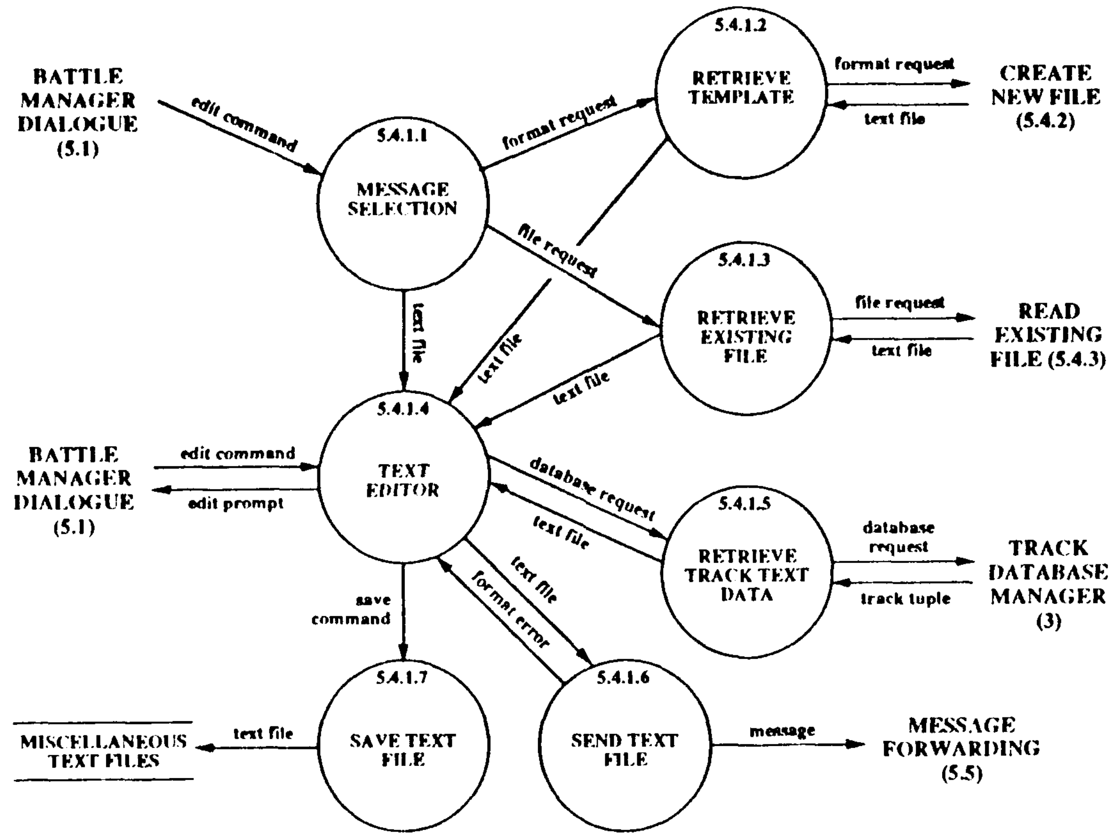
System Output
Expansion of Bubble
5.1.2
WEAPONS SYSTEMS INTERFACE (6)
emergency weapon status report
5.1.2.1 DISPLAY EMERGENCY STATUS REPORT
terminal output (emergency status screen)
User
TEXTUAL MESSAGE DISPLAY (5.6)
new message notice
5.1.2.2 MESSAGE ARRIVAL DISPLAY
terminal output (message arrival notification)
User
PLATFORM STATUS DISPLAY (5.2)
status report
5.1.2.3 DISPLAY WEAPON STATUS
terminal output (weapon status screen)
User
GRAPHICAL TRACK DISPLAY (5.3)
graphics display
5.1.2.4 TRACK DISPLAY
terminal output (graphics display screen)
User
MESSAGE GENERATOR (5.4)
edit prompt
5.1.2.5 DISPLAY EDIT SCREEN
terminal output (edit screen)
User
TEXTUAL MESSAGE DISPLAY (5.6)
text file
5.1.2.6 DISPLAY TEXT FILE
terminal output (text screen)
User



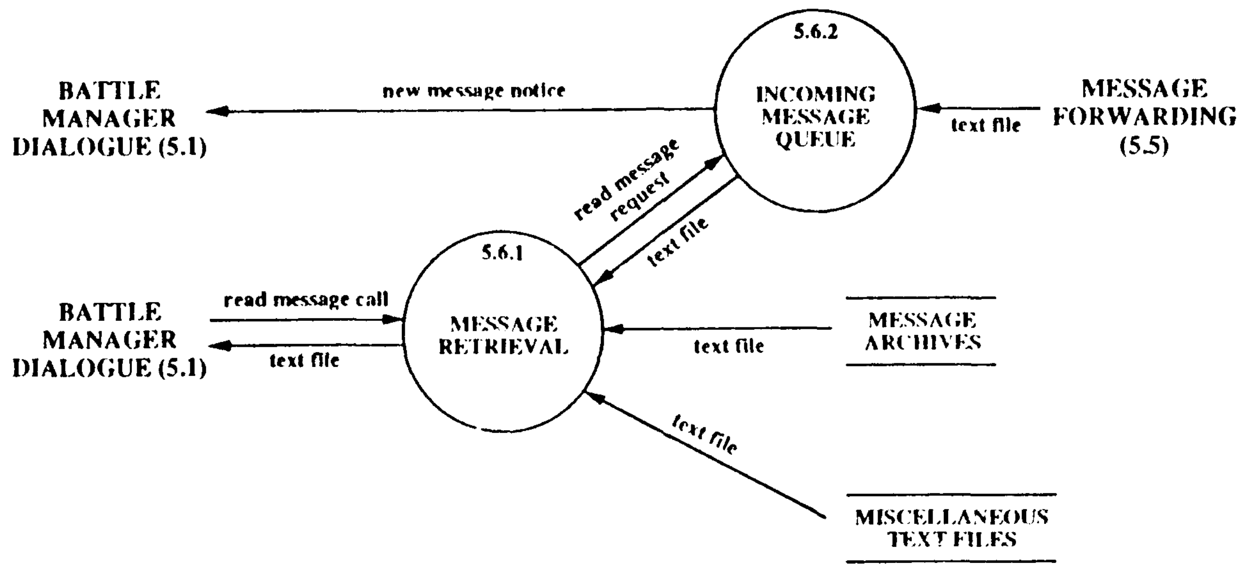

| | |
|---|---|
| **Graphical Track Display** | Expansion of Bubble **5.3** |



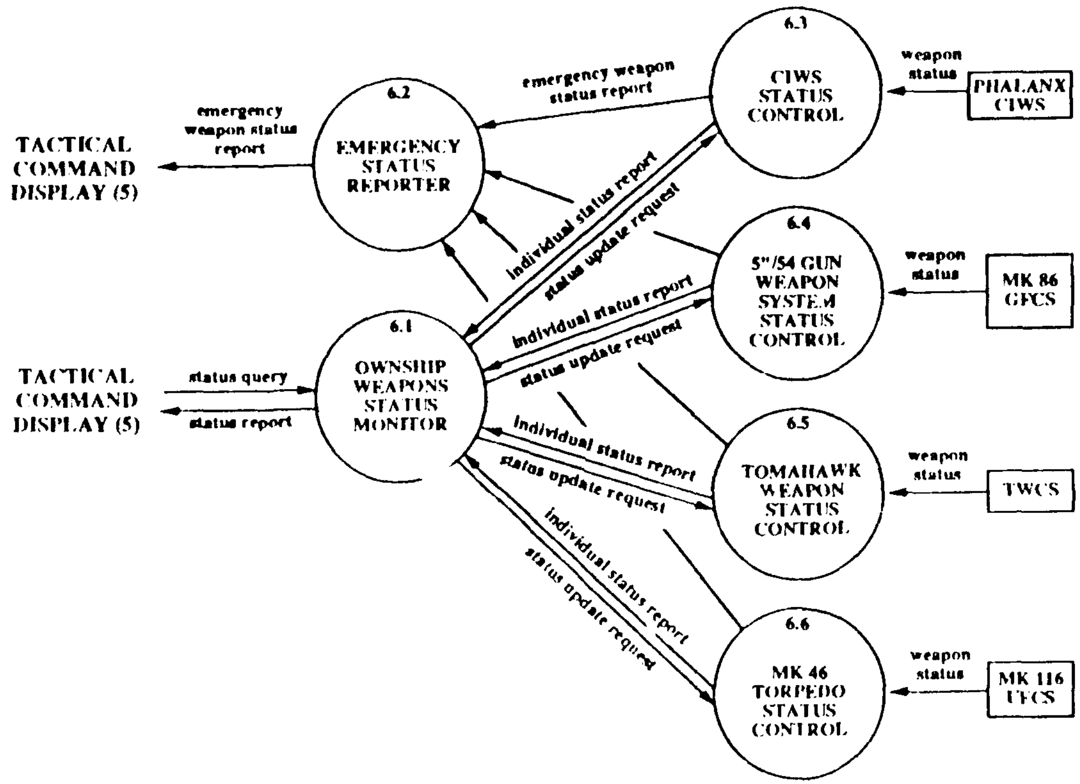

**Message Generator** | Expansion of Bubble **5.4**




| | |
|---|---|
| **Edit Dialogue** | Expansion of Bubble **5.4.1** |




| | |
|---|---|
| **Textual Message Display** | Expansion of Bubble **5.6** |




| | |
|---|---|
| **Weapons Systems Interface** | Expansion of Bubble **6** |

## APPENDIX D

## GENERIC C3I WORKSTATION PROCESS SPECIFICATIONS

# 1 COMMUNICATIONS INTERFACE (ACCEPT, FORMAT & ROUTE)

## 1.1 COMMS MESSAGE CONTROL

### 1.1.1 MESSAGE FORWARDING

The Message Forwarding process shall verify the "parseable" representation of a digital communication message, and act as a preliminary filter for the system.

The Message Forwarding module shall have a maximum execution time of 50 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

> A *local receive order* is received.
> [This is a local system acknowledgement that a message has arrived.]

Precondition 2

> A *transmission command* is received.
> [A textual message is queued to be transmitted.]

Postcondition 1

> When precondition 1 is met, this module shall forward a *reception notification* to Communications Network Monitor and Control (1.4).
> [If the addressee is the same as ownship, this module shall continue no further. Otherwise, this module shall pass the reception notice onto the Communications Network Monitor and Control (1.4), to determine if anything further needs to be done with this message.]

Postcondition 2

When precondition 2 is met, this module shall forward a *local transmit order* to either Link-4a Message Control (1.1.3), Link-11 Message Control (1.1.4), Link-16 Message Control (1.1.5), or OTCIXS Message Control, as specified by the *transmission command*.
[This module shall send the message to the appropriate communications system, along with any additional pertinent data .]

### 1.1.2 INPUT MESSAGE RECEIVER

If a complete, valid and separate communications message/packet is received, then the Input Message Receiver shall send it on for processing.

The Input Message Receiver module shall have a maximum execution time of 50 ms.

This module shall be initiated when the following precondition occurs:

Precondition

A communications message (*text file*) is received.

Postcondition

When the precondition is met, this module shall forward the message (*text file*) onto the Message Librarian (1.2).
[This is the start of the main message processing sequence.]

### 1.1.3 LINK-4A MESSAGE CONTROL

Provided that ownship has Link-4a communications equipment, the Link-4a Message Control process is the workstation interface to that system. This module shall be concerned with the message protocols, word length conversions, encryption protocols (if necessary), and bit patterns necessary to receive and transmit a communications message.

The Link-4a Message Control module shall have a maximum execution time of 50 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

An inbound *communications message* is received.

Precondition 2

A *local transmit order* is received, and an outbound communications message is queued to transmit.

Postcondition 1

When precondition 1 is met, this module shall forward the *local receive order* to the Input Message Receiver (1.1.2).
[If necessary, this module shall turn the message into an ASCII text file, so that the system may manipulate it. Also,this module shall create the message header as described in the Data Dictionary (Appendix E)].

Postcondition 2

When precondition 2 is met, this module shall send the appropriately formatted *communications message* to the communications hardware.
[This module shall make whatever conversions are necessary for the given text file to be transmitted by the communications system.]

### 1.1.4 LINK-11 MESSAGE CONTROL

Provided that ownship has Link-11 equipment, the Link-11 Message Control process is the workstation interface to that system. This module shall be concerned with the message protocols, word length conversions, encryption scheme (if necessary), and bit patterns necessary to receive and transmit a Link-11 message/data packet.

The Link-11 Message Control module shall have a maximum execution time of 50 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

An inbound *communications message* (data packet) is received.

Precondition 2

A *local transmit order* is received, and an outbound communications message (data packet ) is queued to transmit.

Postcondition 1

When precondition 1 is met, this module shall forward the *local receive order* to the Input Message Receiver (1.1.2).
[If necessary, this module shall turn the message into an ASCII text file, so that the system may manipulate it. Also,this module shall create the message header as described in the Data Dictionary (Appendix E)].

Postcondition 2

When precondition 2 is met, this module shall send the appropriately formatted *communications message* to the communications hardware.
[This module shall make whatever conversions are necessary for the given text file to be transmitted by the communications system.]

### 1.1.5 LINK-16 MESSAGE CONTROL

Provided that ownship has Link-16/JTIDS equipment, the Link-16 Message Control process is the workstation interface to that system. This module shall be concerned with the message protocols, word length conversions, encryption commands (if necessary), and bit patterns necessary to receive and transmit a Link-16 message/data packet.

The Link-16 Message Control module shall have a maximum execution time of 50 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

An inbound *communications message* (data packet) is received.

Precondition 2

A *local transmit order* is received, and a message/data packet is queued to transmit (Outbound).

Postcondition 1

When precondition 1 is met, this module shall forward the *local receive order* to the Input Message Receiver (1.1.2). [If necessary, this module shall turn the message into an ASCII text file, so that the system may manipulate it. Also,this module shall create the message header as described in the Data Dictionary (Appendix E)].

Postcondition 2

When precondition 2 is met, this module shall send the appropriately formatted *communications message* to the communications hardware. [This module shall make whatever conversions are necessary for the given text file to be transmitted by the communications system.]

### 1.1.6 OTCIXS MESSAGE CONTROL

Provided that ownship has OTCIXS equipment, the OTCIXS Message Control process is the workstation interface to that system. This module shall be concerned with the message protocols, word length conversions, encryption scheme (if necessary), and bit patterns necessary to receive and transmit a OTH-T Gold message.

The OTCIXS Message Control module shall have a maximum execution time of 50 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

An inbound *communications message* (OTH-T Gold message) is received.

Precondition 2

A *local transmit order* received, and outbound OTH-T Gold message queued to transmit.

Postcondition 1

When precondition 1 is met, this module shall forward *local receive order* to Input Message Receiver (1.1.2).
[If necessary, this module shall turn the message into an ASCII text file, so that the system may use it. Also,this module shall create the message header as described in the Data Dictionary (Appendix E)].

Postcondition 2

When precondition 2 is met, this module shall send the appropriately formatted *communications message* to the communications hardware.
[This module shall make whatever conversions are necessary for the given text file to be transmitted by the communications system.]

## 1.2 MESSAGE LIBRARIAN

### 1.2.1 ARCHIVE FILTER

The purpose of the Archive Filter process is to limit the number of inbound messages to be processed by the system. Prior to initiation, filter constraints shall be set by the user to indicate which messages are to be accepted by the system (by type, time-late, classification, addressee, etc.).

This module shall also limit the number of outbound messages that are to be saved in the Message Archives.

The Archive Filter module shall have a maximum execution time of 50 ms.

This module shall be initiated when any of the following preconditions are met:

Precondition 1

An *archive setup* command is received.

Precondition 2

An inbound or outbound message (*text file*) is received, and message violates archive constraints.

Precondition 3

An inbound or outbound message (*text file*) is received, and archive constraints are met.

Postcondition 1

When precondition 1 is met, this module shall set local archive constraints according to user input.
[Local archive constraints determine what messages are to be processed, what messages are to be archived and what messages get pigeon holed.]

Postcondition 2

When precondition 2 is met, this module shall not archive the message.

Postcondition 3

When precondition 3 is met, this module shall forward the *text file* to Save Message Text (1.2.2).

### 1.2.2 SAVE MESSAGE TEXT

The Save Message Text process saves a copy of the given message (as a text file) in memory. Text file names are to be unique identifiers.

This module shall also download "non-current" files onto a mass storage device as is applicable, or deemed necessary by system constraints.

The Save Message Text module shall have a maximum execution time of 100 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

An inbound *message* is received.

Precondition 2

An outbound *message* is received.

Postcondition 1

When precondition 1 is met, this module shall save the *message* as a *text file* in Message Archives, and forward the *message* to Track-Text Sorter (1.2.3)

Postcondition 2

When precondition 2 is met, this module shall save the *message* as a *text file* in Message Archives.

### 1.2.3 TRACK-TEXT SORTER

The Track-Text Sorter process shall determine whether the message contains track information, textual information or both.

The Track-Text Sorter module shall have a maximum execution time of 100 ms.

This module shall be initiated when any of the following preconditions are met:

Precondition 1

A *message* is received and it contains textual information only.

Precondition 2

A *message* is received and it contains track information only.

Precondition 3

A *message* is received and it contains textual and track information.

Postcondition 1

When precondition 1 is met, this module shall forward the *message* (analogous to *electronic mail*) to be read by the user.

Postcondition 2

When precondition 2 is met, this module shall forward the *message* on to the Track Extractor (1.3).

Postcondition 3

When precondition 3 is met, this module shall forward the *message* ("*electronic mail*") to be read by the user, and forward the *message* on to the Track Extractor (1.3).

### 1.3 TRACT EXTRACTOR

### 1.3.1 TRACK-CONTACT SORTER

The Track-Contact Sorter process shall parse through the given message and remove all information that is not relevant to track reports.

The Track-Contact Sorter module shall have a maximum execution time of 50 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A *message* containing specific track/target information is received. [Track/target information includes specific track identification, position, lat/long, bearing/range, etc.]

Precondition 2

A *message* containing specific contact reports/probable target information is received. [Contact report information may consist of an ESM report, jamming strobe, or a limited subset of track information (e.g., azimuth information only).]

Postcondition 1

When precondition 1 is met, this module shall forward the appropriate *track line* to the Comms Track Synthesis (1.3.2).

Postcondition 2

When precondition 2 is met, this module shall forward appropriate *contact line* to Comms Contact Synthesis (1.3.3).

### 1.3.2 COMMS TRACK SYNTHESIS

The Comms Track Synthesis process shall parse through the given track lines (a subset of a communications message), and create a track tuple which contains all appropriate and/or known information about the given track.

The Comms Track Synthesis module shall have a maximum execution time of 25 ms.

This module shall be initiated when the following precondition is met:

Precondition

*Track lines* are received.

Postcondition

When the precondition is met, this module shall create a corresponding *track tuple* , and place it in the Tuple Forwarding queue (1.3.4).

### 1.3.3 COMMS CONTACT SYNTHESIS

The Comms Contact Synthesis process shall parse through the given contact lines (a subset of a communications message), and create a track tuple which contains all appropriate and/or known information about the given contact/track.

The Comms Contact Synthesis module shall have a maximum execution time of 25 ms.

This module shall be initiated when the following precondition is met:

Precondition

*Contact lines* are received.

Postcondition

When the precondition is met, this module shall create a corresponding *track tuple* , and place it in the Tuple Forwarding queue (1.3.4).

### 1.3.4 TUPLE FORWARDING

The Tuple Forwarding process shall prioritize and queue track tuples. While any given implementation could easily alter the priority scheme presented here, the following precedence scheme is merely an example (higher priorities come first):

- air tracks
- surface tracks
- subsurface tracks

Within these demarcations, additional priorities could include:

- fastest tracks
- most recent tracks [smallest time-late]
- hostile tracks
- neutral/unknown tracks
- friendly tracks
- ingressing tracks (optional)

The Tuple Forwarding module shall have a maximum execution time of 25 ms.

This module shall be initiated when the following precondition is met:

Precondition

*Track tu s* are received.

Postcondition

When the precondition is met, this module shall forward highest priority *track tuples* (in sequence) to Track Database Manager (3).

## 1.4 COMMUNICATIONS NETWORK MONITOR & CONTROL

### 1.4.1 INBOUND MESSAGE PROCESSING

The Inbound Message Processing process shall contain the intelligence necessary to act as a Master (Controlling) Unit in a communications network,. In order to do this well, this module could itself be expanded into an expert system.

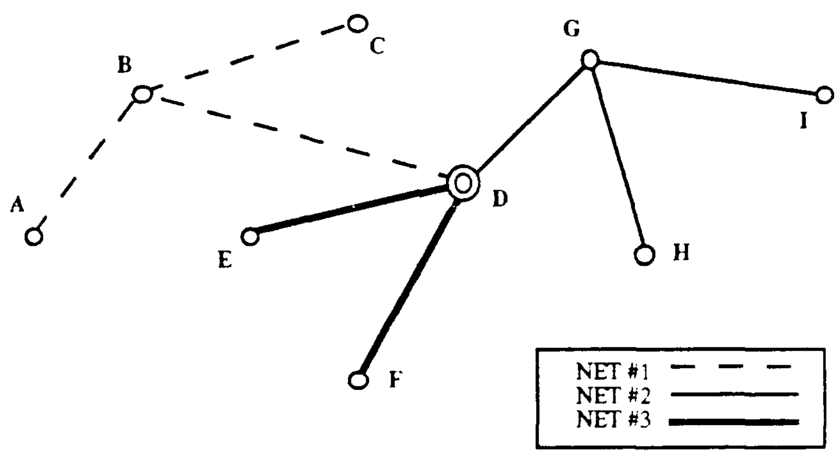



**Figure D-1. Example Network**

Information shall be needed concerning who are on a particular network. The "network setup" message would contain this information (cf. Figures D-1 and D-2.). However, intelligent on-line programming would enable the system to monitor inbound messages and deduce network participants.

**Link #1, Channel A**
Platform A via B
Platform B
Platform C via B
Platform D

**Link #2, Channel B**
Platform D
Platform E
Platform F

**Link #3, Channel C**
Platform D
Platform G
Platform H via G
Platform I via G

**Figure D-2. Example network setup message (based upon Figure D-1)**

The Inbound Message Processing module shall have a maximum execution time of 500 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A *network setup* command is received.

Precondition 2

A *reception notification* (new inbound message) is received.

Postcondition 1

When precondition 1 is met, this module shall set internal variables. [Which units are participating units in a given network? What are special considerations necessary for interconnecting networks?]

Postcondition 2

When precondition 2 is met, this module shall parse through communications message, and verify to whom it is addressed, from whom it was sent to determine if this message needs to be relayed. If appropriate, forward *relay command* to the Outbound Message Processing (1.4.2).

## 1.4.2 OUTBOUND MESSAGE PROCESSING

The Outbound Message Processing process shall verify that a given message format is acceptable for transmission over a specified link. If the link is not specified, then the message shall be parsed in order to determine if the link may be inferred (e.g., OTH-T Gold messages implicitly suggest transmission over OTCIXS).

The Outbound Message Processing module shall have a maximum execution time of 300 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A *relay command* is received from the Inbound Message Processor (1.4.1).

Precondition 2

A *transmit command* is received from the Tactical Command Display (5).

Postcondition 1

When precondition 1 is met, this module shall translate the message if necessary. This module shall send the approved message to Transmission Forwarding (1.4.4).

Postcondition 2

When precondition 2 is met, this module shall send a *translation command* to the Translation Interface (1.4.3). This module shall forward the translated message (*text file*) to the Message Librarian (1.2). This module shall also send the *transmission command* to Transmission Forwarding (1.4.4).

## 1.4.3 TRANSLATION INTERFACE

It is assumed that message format translation may be a CPU intensive operation. Hence, the Translation Interface process shall queue format translation requests, and forward the modified messages back to Outbound Message Processing (1.4.2).

The Translation Interface module shall have a maximum execution time of 100 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A *translation command* is received.
[The current format and the desired format must be specified.]

Precondition 2

A translated *text file* is received.
[A successful translation has occurred.]

Postcondition 1

When precondition 1 is met, this module shall prioritize and forward the *translation commands* to the Format Translator (1.5).

Postcondition 2

When precondition 2 is met, this module shall translate and forward the *text file* back to Outbound Message Processing (1.4.2).

### 1.4.4 TRANSMISSION FORWARDING

The Transmission Forwarding process shall prioritize and queue up all messages to be transmitted.

The Transmission Forwarding module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions are met:

Precondition 1

An *emissions control command* is received.
[Permission to transmit is either granted or denied.]

Precondition 2

A *transmission command* is received.

Precondition 3

A *periodic transmission command* is received.

Postcondition 1

When precondition 1 is met, if an EMCON order is issued, this module shall cease and desist transmissions.
[This module shall save messages in queue in local buffer for recovery. This module shall also remove all messages from the transmit queue.]

Otherwise, this module shall resume forwarding *transmission commands* to Comms Message Control (1.1). [This module shall download messages from local buffer into the transmit queue and resume processing.]

Postcondition 2

When precondition 2 is met, this module shall evaluate the message priority and enter *transmission commands* into the transmit queue.

Postcondition 3

When precondition 3 is met, this module shall evaluate the message priority and enter *transmission commands* into the transmit queue.

## 1.5 FORMAT TRANSLATOR

While it is easy to state that a module will need to be created that is capable of changing textual messages from one format standard into another, it is quite another to implement it. Indeed, the Format Translator process must be exceptionally intelligent in order to perform its task well -- quickly, and accurately.

The Format Translator module shall have a maximum response time of 300 ms.

This module shall be initiated when the following precondition is met:

Precondition

A *translation command* queued.
Format translation conversion is possible.

Postcondition

When the precondition is met, this module shall forward the translated *text file* to Communication Network Monitor & Control (1.4).

## 1.6 PERIODIC TRANSMISSION GENERATOR

### 1.6.1 PERIODIC REPORT GENERATOR

The Periodic Report Generator shall originate all routine messages of a periodic nature (e.g., Link-11 track reports, or OTH-T Gold SITREPs).

The Periodic Report Generator module shall execute at a user defined interval and shall have a maximum response time of 500 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

An *initiate transmission sequence command* is received.

Precondition 2

Specified timing intervals are met.

Postcondition 1

When precondition 1 is met, this module shall Initialize internal variables including message type, frequency the message is to be generated, and information included in the message.

Postcondition 2

When precondition 2 is met, this module shall retrieve appropriately formatted *message template*, retrieve appropriate *report data*, and forward the *periodic transmission command* on to Communications Network Monitor and Control (1.4).

### 1.6.2 TRACK REPORT

Provided that a periodic message requires information pertaining to tracks, or the track database, then the Track Report process shall fetch that information, and return it to the Track Database Manager in a text file format.

The Track Report module shall have a maximum execution time of 50 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A *database request* is received.

Precondition 2

*Track tuple* information is received.

Postcondition 1

When precondition 1 is met, this module shall forward a *database request* to Track Database Manager (3).

Postcondition 2

When precondition 2 is met, this module shall parse the track information (*report data*) and place it into an ASCII text file format that is acceptable for the given message format.

### 1.6.3 MESSAGE FORMAT TEMPLATE

The Message Format Template process shall maintain a local library of message format templates and shall return a specific template upon request.

The Message Format Template module shall have a maximum response time of 50 ms.

This module shall be initiated when the following precondition is met:

Precondition

A *desired format* for a message template is requested.

Postcondition

When the precondition is met, this module shall return a *Message template* to the Periodic Report Generator (1.6.1).

# 2 SENSOR INTERFACE (ACCEPT & FORMAT)

## 2.1 SENSOR INTERFACE CONTROL

### 2.1.1 SENSOR CONTACT SYNTHESIS

The Sensor Contact Synthesis process shall act as a track filter, eliminating duplicate or redundant information. If a given target is being tracked by more than one sensor, this module must decide which sensor is providing more accurate and timely information, or combine the sensor information into a single track.

It must be assumed that sensors will not be synchronized, and contact reports (even of the identical target) will not be reported simultaneously. This module shall maintain a data store that keeps a history of most recently reported contacts. Based upon this data store, contact reports from differing sensors shall be compared.

Further, older sensor systems designed and built by Raytheon, RCA, General Electric, etc. produced information that was never intended to be match, merged, and correlated with other systems. Track identification and naming conventions must be assumed to vary. This module must also insure the consistency and uniqueness of track identification and labeling.

The Sensor Contact Synthesis module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

Sensor *contact data* are received.

Postcondition

When the precondition is met, this module shall forward *local track information* to Sensor Information Normalization (2.2).

### 2.1.2 INTELLIGENCE SYNTHESIS

Information that cannot be directly correlated with a specific track may still be of tactical importance. Certain sensors, such as a passive electronic support measures (ESM) device (e.g., SLQ-32), may provide information on the transmission frequencies of a radar. Based on intelligence information, it may be possible to deduce what type of radar operates within those bandwidths, and whether that radar is airborne, surface mounted, etc..

The Intelligence Synthesis process may include an expert system of a Tactical Action Officer, and provide decision support concerning sensor intelligence information.

The Intelligence Synthesis module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition 1

*Intelligence data* are received.

Postcondition 1

When precondition 1 is met, this module shall store the *Intelligence data*. When *intelligence data* are correlated, this module shall forward the *Intelligence report* to the Track Controller (4).

### 2.1.3 RADAR INTERFACE CONTROL

Provided that ownship is equipped with a radar, the Radar Interface Control process is the workstation interface to the radar system. This module shall be concerned with the message protocols and bit patterns necessary to retrieve track information from the radar system, and shall translate that information into a standardized contact report. Any relevant track information that can be provided by the given radar shall be extracted from the radar data.

N.B. Any platform equipped with more than one radar would require more than one Radar Interface Control module.

The Radar Interface Control module shall have a maximum execution time of 30 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A radar track (*sensor information*) is received.
[Again, this information is radar system dependent and could potentially be 2D or 3D,.Doppler, etc.]

Precondition 2 (Optional)

A radar derived *intelligence data* is received.
[For those radars, such as inverse synthetic aperture radars (ISAR), that are capable of providing unique target specific information, this module shall gather and pass this information along for intelligence purposes.]

Postcondition 1

When precondition 1 is met, this module shall put the *contact data* into standardized format and forward it to Sensor Contact Synthesis (2.1.1).

Postcondition 2 (Optional)

When precondition 2 is met, this module shall forward the *intelligence data* to Intelligence Synthesis (2.1.2). Intelligence information that is directly correlatable to a specific contact shall also be placed into the *contact data* message.

### 2.1.4 SONAR INTERFACE CONTROL

Provided that ownship is equipped with a sonar, the Sonar Interface Control process is the workstation interface to the sonar system. This module shall be concerned with the message protocols and bit patterns necessary to retrieve track information from the sonar system, and shall translate that information into a standardized contact report. Any relevant track information that can be provided by the given sonar will be extracted from the acoustic data.

N.B. Any platform equipped with more than one sonar (active bow mounted, and passive towed array sonar) would require more than one Sonar Interface Control module.

The Sonar Interface Control module shall have a maximum execution time of 30 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

Sonar contact data (*sensor information*) are received.
[Again, this information is sonar system dependent and could potentially be 2D or 3D,.Doppler, etc.]

Precondition 2 (Optional)

Sonar derived *intelligence data* are received.
[For those sonars that are capable of providing unique target specific information (e.g., acoustic & cavitation specific data), then this module shall gather and pass this information along for intelligence purposes.]

Postcondition 1

When precondition 1 is met, this module shall put the *contact data* into standardized format and forward it to Sensor Contact Synthesis (2.1.1).

Postcondition 2 (Optional)

When precondition 2 is met, this module shall forward the *intelligence data* to Intelligence Synthesis (2.1.2). Intelligence information that is directly correlatable to a specific contact shall also be placed into the *contact data* message.

### 2.1.5 INFRARED INTERFACE CONTROL

Provided that ownship is equipped with an infrared search and track (IRST) sensor system, the Infrared Interface Control process is the workstation interface to the IRST system. This module shall be concerned with the message protocols and bit patterns necessary to retrieve track information from the IRST system, and shall translate that information into a standardized contact report. Any relevant track information that can be provided by the given IRST shall be extracted from the infrared data.

N.B. Any platform equipped with more than one IRST would require more than one Infrared Interface Control module.

The Infrared Interface Control module shall have a maximum execution time of 30 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

IRST track data (*sensor information*) are received.

Precondition 2 (Optional)

Infrared sensor derived *intelligence data* are received.
[For those infrared sensors capable of providing unique target specific information (e.g., number of stacks or jet engines), then this module shall gather and pass along this information for intelligence purposes.]

Postcondition 1

When precondition 1 is met, this module shall put the *contact data* into standardized format and forward it to Sensor Contact Synthesis (2.1.1).

Postcondition 2 (Optional)

When precondition 2 is met, this module shall forward the *intelligence data* to Intelligence Synthesis (2.1.2). Intelligence information that is directly correlatable to a specific contact shall also be placed into the *contact data* message.

### 2.1.6 ESM INTERFACE CONTROL

Provided that ownship is equipped with a ESM device, the ESM Interface Control process shall be the workstation interface to the ESM system. This module shall be concerned with the message protocols and bit patterns necessary to retrieve track information from the radar system, and shall translate that information into a standardized contact report. Any relevant track information that can be provided by the given ESM device will be extracted from the electromagnetic spectrum data.

N.B. Any platform equipped with more than one ESM device would require more than one ESM Interface Control modules.

The ESM Interface Control module shall have a maximum execution time of 30 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

An ESM track (*sensor information*) is received.
[Again, this information is system dependent and could include directional information without range, direction and range, direction and elevation, etc.]

Precondition 2

*Intelligence data* are received.
[All intercepted electromagnetic emissions shall be collected and passed along for intelligence purposes.]

Postcondition 1

When precondition 1 is met, this module shall put the *contact data* into standardized format and forward it to Sensor Contact Synthesis (2.1.1).

Postcondition 2

When precondition 2 is met, this module shall forward the *intelligence data* to Intelligence Synthesis (2.1.2). Intelligence information that is directly correlatable to a specific contact shall also be placed into the *contact data* message.

## 2.2 SENSOR INFORMATION NORMALIZATION

Sensors do not occupy the same location on a platform. They will be located anywhere from a few meters apart, to a hundred meters apart, or more. The Sensor Information Normalization process shall take into account the physical location of a reporting sensor, and modify its reporting information so that it uses the identical reference point as all other sensors.

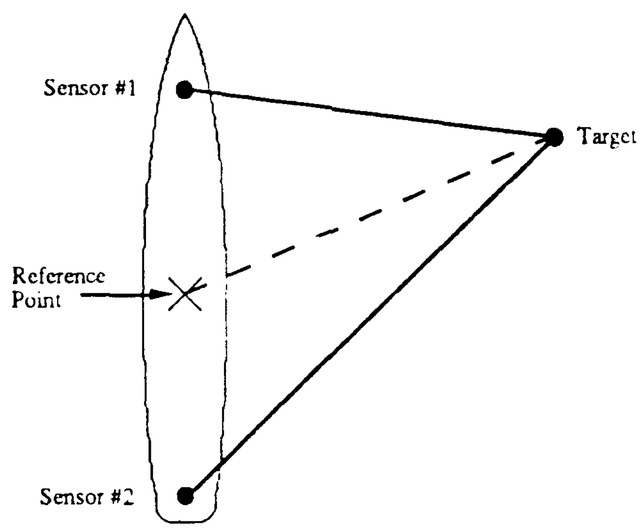



**Figure D-3. Platform Reference Point**

NOTE: Many sensor systems already perform this function, and report their information from a platform specific reference point.

The Sensor Information Normalization module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Local track information* is received.

Postcondition

When the precondition is met, this module shall adjust angular measurements to one point on ownship and forward the updated *local track information* to Relative-to-Absolute Position Conversion (2.4).

## 2.3 OWNSHIP LOCATION MONITOR

The Ownship Location Monitor process interfaces with the ownship navigation system. The ownship latitude, longitude, course, velocity and current time are needed for turning relative track data into absolute (in terms of global references -- latitude and longitude). This module shall be triggered by navigation system inputs and must operate under hard real-time constraints, in order to provide accurate positioning information.

It may well be that for certain implementations, this information may require interfacing with multiple systems. However, it is anticipated that the advent of the Global Positioning System (GPS) in the mid- or late-1990's will provide a common standardize navigation system.

Provided that "ownship" is, in fact, a mobile platform, and is equipped with a navigation system (or set of navigation systems), this is the workstation interface to the navigation system(s). This module would be concerned with the message protocols and bit patterns necessary to retrieve ownship position information from the navigation system(s), and translate that information into a standardized format.

The Ownship Location Monitor module shall have a maximum response time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

A navigation system information update (*ownship navigation information*) is received and a significant update of ownship latitude, longitude, course, velocity or time has occurred.[The frequency of positional information update will be a function of "ownship" speed. The faster the motion, the more frequentlythe information should be updated.]

Postcondition

When the precondition is met, this module shall forward *ownship navigation information* to Relative-to-Absolute Position Conversion (2.4).

## 2.4 RELATIVE-TO-ABSOLUTE POSITION CONVERSION

The Relative-to-Absolute Position Conversion process must take in relative positional information (bearing and range from ownship), and convert this information into absolute positional information (latitude, longitude, etc.).

The Relative-to-Absolute Position Conversion module shall have a maximum execution time of 50 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

An update of *ownship navigation information* is received.

Precondition 2

*Local track information* is received.

Postcondition 1

When precondition 1 is met, this module shall update local variables for calculating relative-to-absolute position conversion.

Postcondition 2

When precondition 2 is met, this module shall convert relative elevation, azimuth and range into latitude and longitude and forward the *track tuple* information to Track Database Manager (3).

# 3 TRACK DATABASE MANAGER

## 3.1 TRACK DATABASE UPDATE

### 3.1.1 DATABASE MESSAGE CONTROL

Any request to add a new track, modify or delete an existing track shall be handled by the Database Message Control process.

In the interests of computer security, this module may be modified to perform an authorization validation prior to altering the track database.

The Database Message Control module shall have a maximum execution time of 10 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

A *add track tuple* to track database request is received.

Precondition 2

A *delete track tuple* from track database request is received.

Precondition 3

A *update track tuple* in track database request is received.

Postcondition 1

When precondition 1 is met, this module shall forward the *add track tuple* request on to the Database Filter (3.1.2).

Postcondition 2

When precondition 2 is met, this module shall forward the *delete track tuple* request on to Remove Track Tuple (3.1.5).

Postcondition 3

When precondition 3 is met, this module shall forward the *update track tuple* request on to Change Attribute Value (3.1.6).

### 3.1.2 DATABASE FILTER

The Database Filter process shall provide the user a robust means of preventing unwanted tracks from entering the track database. A track may be filtered from entering the database by comparing any of its attributes with those of interest.

If the user is only interested in tracks within 1000 nautical mile range, then any tracks reported outside this region shall be kept from entering the track database. If the user is interested in surface tracks within the Persian Gulf as well as any air tracks within 500 nautical miles, then this module may cross reference range versus track type. If a user were only interested in contacts within the Caribbean Sea, a reasonably complex geographic check could be made to verify that a track, although in close proximity, was not on the Pacific side of the Isthmus of Panama. Further, commercial fishing vessels may be filtered from entering the database.

The Database Filter module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

A *set track filter* command is received.

Precondition 2

An *add track tuple* request is received.
The track tuple passes filter requirements, and is considered wanted.

Precondition 3

An *add track tuple* request is received.
The track tuple fails filter requirement, and is considered unwanted.

Postcondition 1

When precondition 1 is met, this module shall update local variables for filtering out unwanted tracks.

Postcondition 2

When precondition 2 is met, this module shall forward *track tuple* to the Prioritize Tuple queue (3.1.3).

Postcondition 3

When precondition 3 is met, this module shall do nothing.

### 3.1.3 PRIORITIZE TUPLES

The Prioritize Tuples process serves as an additional track queue (cf. Process 1.3.4). Again, track tuples are prioritized and queued. While any given implementation could easily alter the priority scheme presented here, the following precedence is merely provided for the sake of example (higher priorities come first):

- air tracks
- surface tracks
- subsurface tracks

Within these demarcations, additional priorities could include:

- fastest tracks
- most recent tracks [smallest time-late]
- hostile tracks
- neutral/unknown tracks
- friendly tracks
- ingressing tracks (optional)

N.B., the priority scheme presented here reiterates that of Process 1.3.4, they may differ. Under certain circumstances it may be best to have these precedences vary slightly.

The Prioritize Tuples module shall have a maximum execution time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Track tuples* are received.

Postcondition

When the precondition is met, this module shall forward the highest priority *track tuple* (in sequence) to Write Tuple to Database (3.1.4).

### 3.1.4 WRITE TUPLE TO DATABASE

The Write Tuple To Database process shall modify the track database. This module shall also resolve memory allocation conflicts within the track database. If the track database is designed to operate with no more than $x$ track tuples, then the moment tuple number $x+1$ attempts to be added to the database this function shall choose one track to drop (likely by using the reverse order of the precedences presented in Process 3.1.3).

If a segmented memory scheme is adopted -- memory allocated for $x$ air tracks, $y$ surface tracks, and $z$ subsurface tracks -- then provided that the $x+1$ air track attempts to be added to the database, this module may conceivably dynamically alter the bounds of subsurface tracks to $z-1$ (for a limited period of time, and provided that there are not $z$ subsurface tracks within the database at the time).

The Write Tuple To Database module shall have a maximum execution time of 500 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A *track tuple* is queued to be added to the track database, and there is sufficient memory allocated for the inclusion of the track tuple.

Precondition 2

A *track tuple* is queued to be added to the track database, and there is not sufficient memory allocated for the inclusion of the track tuple.

Postcondition 1

When precondition 1 is met, this module shall add the *Track tuple* to the track database.

Postcondition 2

When precondition 2 is met, this module shall remove the least significant *track tuple* from the database, and the new *track tuple* is added to the track database. Or, provided that the new track is less significant than any extant track, then this module shall not add the new *track tuple* into the track database.

### 3.1.5 REMOVE TRACK TUPLE

The Remove Track Tuple process shall queue delete track tuple requests, and remove the specified tracks from the track database.

The Remove Track Tuple module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

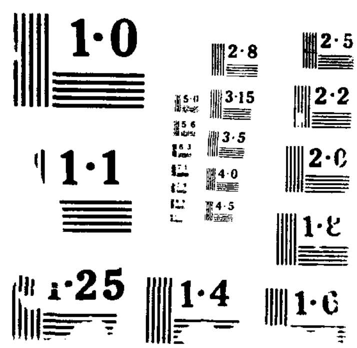
Precondition

A *delete track tuple* request is received.

AD-A241 377   FUNCTIONAL SPECIFICATION FOR A GENERIC C3I WORKSTATION   3/3
(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA
S E ANDERSON SEP 90 XN-NPS

UNCLASSIFIED   NL

END
FILMED
DTIC


1·0
2·8
2·5
3·15
2·2
3·5
1·1
2·0
4·0
4·5
1·25
1·4
1·6

Postcondition

When the precondition is met, this module shall remove the *track tuple* with the specified *track ID* from the track database.

### 3.1.6 CHANGE ATTRIBUTE VALUE

The Change Attribute Value process shall queue update track tuple requests, and modify the specified tracks in the track database.

The Change Attribute Value module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition

An *update track tuple* request is received.

Postcondition

When the precondition is met, this module shall modify the specified *track tuple* within the track database.

## 3.2 TRACK REQUEST

### 3.2.1 MANAGE TRACK DATABASE REQUEST

Any request to read information from the track database shall be first processed by the Manage Track Database Request process. This module shall queue database requests. (Requests could be prioritized according the the requesting party.)

This module may also perform authorization checks in order to insure that the information returned does not exceed the classification of the requesting party.

The Manage Track Database Request module shall have a maximum execution time of 50 ms.

This module shall be initiated when the following precondition occurs:

Precondition

A *Database request* is received and queued.
This module shall verify and validate the request .

Postcondition

When the precondition is met, this module shall forward the *Database request* to Access Track Tuple (3.2.2).

### 3.2.2 ACCESS TRACK TUPLE

The Access Track Tuple process shall retrieve the actual data that satisfies the track database request. The database information shall then be sent to Forward Track Tuple (3.2.3) along with information about the module that made the request.

The specific implementation of how the track data is stored, and what data storage retrieval methods are to be employed are open issues. Should the database be implemented using the relational data model (RDM), then the data requests would need to be placed in a system query language (SQL) format. If, however, the track database uses an object-oriented data model (OODM), then a compatible query language would need to be used. (At the time of this writing, commercially available object-oriented database systems are in their infancy.)

The Access Track Tuple module shall have a maximum response time of 900 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

A *Database request* is received for current track data.

Precondition 2

A *database request* is received for non-current track data

Precondition 3

A *Database request* is received for both current and non-current track data.

Postcondition 1

When precondition 1 is met, this module shall retrieve Pertinent *track tuple* information from the Track Database. This information shall be passed on, with the name of the module that made the request (*origin*), to Forward Track Tuple (3.2.3).

Postcondition 2

When precondition 2 is met, this module shall retrieve Pertinent *track tuple* information from the Track Archive database. This information shall be passed on, with the name of the module that made the request (*origin*), to Forward Track Tuple (3.2.3).

Postcondition 3

When precondition 3 is met, this module shall retrieve Pertinent track tuple information from both the Track Database and the Track Archive database. The combined data is forwarded. This information shall be passed on, with the name of the module that made the request (*origin*), to Forward Track Tuple (3.2.3).

### 3.2.3 FORWARD TRACK TUPLE

The Forward Track Tuple process shall return track tuple information to the module that requested the given data.

The Forward Track Tuple module shall have a maximum execution time of 50 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Track tuple* data are received along with the name of the *origin* of the request.

Postcondition

When the precondition is met, this module shall forward the *track tuple* information to the requesting module.

## 3.3 DATABASE MONITOR

### 3.3.1 SCAN TRACK DATABASE

The Scan Track Database process shall periodically scan the track database with the goal of removing old, unwanted, and redundant tracks.

The Scan Track Database module shall execute periodically (as defined by the user) with a maximum execution time of 100 ms. The execution interval defined by the user must be greater than 100 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A *set refresh rate* command is received.

Precondition 2

Refresh time interval has elapsed.

Postcondition 1

When precondition 1 is met, this module shall set the local timing variable to the new refresh rate value.

Postcondition 2

When precondition 2 is met, this module shall copy (download) current track database information, and forward relevant *track tuple* data to the following modules: Timeout (3.3.2), Constraint Violated (3.3.4), and Identify Similarities (3.3.5).

### 3.3.2 TIMEOUT

Track information is considered to be perishable data. Once track information exceeds a certain age, it becomes less valuable, and the Timeout process shall remove the tuple from the current track database and add the tuple in to the Track Archive database.

The Timeout module shall have a maximum execution time of 300 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A *set archive timeout* command is received.

Precondition 2

*Track tuples* are received

Postcondition 1

When precondition 1 is met, this module shall set local variables to the specified values. [These variables shall specify when a given track tuple may be removed due to its age.]

Postcondition 2

When precondition 2 is met, this module shall scan track tuples.
If the track tuple archive flag is set,
then this module shall send the *track tuple* to Archive Tracks (3.3.3) and *delete track tuple* from the track database.
If track tuple violates timing constraints,
then this module shall update the track tuple archive flag and send *update track tuple* to Modify Track Database (3.3.6).

### 3.3.3 ARCHIVE TRACKS

The Archive Tracks process shall archive track data that is no longer current. The ultimate goal shall be to maintain track archives for all tracks ever reported. (This sort of track archives would have been helpful during the Persian Gulf Crisis).

At some point, the quantity of tracks will exceed any system memory constraints, and shall be stored off onto secondary memory storage devices. However, there are many times that non-current tracks may still be displayed (e.g., the last reported position of the Kirov battle cruiser).

If secondary mass storage devices provide rapid data retrieval, then it may be sufficient for the track data to be stored directly onto the secondary storage device. However, if the access delay time is unacceptably long, then a separate track archive database may be maintained by the system, and at periodic intervals this information would be stored onto external mass storage devices.

The Archive Tracks module shall have a maximum execution time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Track tuples* are received and queued to be archived.

Postcondition

When the precondition is met, this module shall store the *Track tuple* in the track archive database.

### 3.3.4 CONSTRAINT VIOLATED

Tracks move. (Ownship may move.) Once track information moves far enough away from ownship to be of interest, then this track shall be removed from the database by the Constraint Violated process.

The Constraint Violated module shall have a maximum execution time of 300 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A *set monitor range* command is received.

Precondition 2

*Track tuples* are received

Postcondition 1

When precondition 1 is met, this module shall set local variables to the specified values. [These variables shall specify when a given track tuple may be removed due to its geographical distance away from ownship.]

Postcondition 2

When precondition 2 is met, this module shall scan track tuples.
If the track tuple exceeds distance constraints,
then this module shall *delete track tuple* from track database.

### 3.3.5 IDENTIFY SIMILARITIES

The Identify Similarities process shall always be on guard against the possibility of the same track being reported by more than one source, and hence being entered into the track database more than once.

This module may consist of an expert system, that mimics the ambiguity resolution functions of a human operator in making decisions which of two tracks are the same. Further, the expert system must be capable of deciding how to correlate, match and merge the information contained in the two track tuples.

The Identify Similarities module shall have a maximum execution time of 700 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A *set monitor mode* command is received.
[This module may be in any of three modes: automatic, advise, off.]

Precondition 2

*Track tuples* are received

Postcondition 1

When precondition 1 is met, this module shall set local variables to the specified values.

Postcondition 2

When precondition 2 is met, this module shall scan track tuples.
If two or more track tuples appear similar, and module is in automatic mode,
then this module may *add track tuple*, or *delete track tuple* from track database or *update track tuple(s)* to resolve ambiguity.
If two or more track tuples appear similar, and module is in advise mode,
then this module shall send a *resolution notice* to the user of the possibility of a track ambiguity.
If module is in off mode,
then this module shall do nothing.

### 3.3.6 MODIFY TRACK DATABASE

After the track database has been scanned to determine if there are any "out of date" tracks, any tracks are beyond the ranges of interest, and any tracks that are redundant, then the Modify Track Database process shall queue the delete or modify track commands.

The Modify Track Database module shall have a maximum execution time of 50 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

A *delete track tuple* request is received.
[Track tuple is no longer current, and has been archived.]
[Track tuple is no longer within range, and should be eliminated.]
[Track tuple is redundant, and should be eliminated.]

Precondition 2

An *update track tuple* request is received.
[Track tuple is no longer current, current flag should reflect this.]
[Track tuple information has been matched, merged, and correlated with another track, and the track database should include these changes.]

Precondition 3

An *add track tuple* request is received.
[New track tuple is created to be added to the database.]

Postcondition 1

When precondition 1 is met, this module shall forward the *delete track tuple* request to Track Database Update (3.1).

Postcondition 2

When precondition 2 is met, this module shall forward the *update track tuple* request to Track Database Update (3.1).

Postcondition 3

When precondition 3 is met, this module shall forward the *add track tuple* request to Track Database Update (3.1).

### 3.3.7 MONITOR SETUP

Since the process of monitoring the track database is a complex process, there will need to be a large number of parameters that will need to be set (or reset) in order to meet the requirements of a given C3I workstation installation. The Monitor Setup process shall take the parameters specified by the track database manager and pass them along to the lower level modules.

The Monitor Setup module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

A *set monitor constraints* command is received.

Postcondition

When the precondition is met, this module shall forward *set refresh rate* command to Scan Track Database (3.3.1), forward *set archive timeout* command.to Timeout (3.3.2), forward *set monitor range* command to Constraint Violated (3.3.4), and forward *set monitor mode* command to Identify Similarities (3.3.5).

## 3.4 OWNSHIP TRACK MONITOR

### 3.4.1 OWNSHIP NAVIGATION MONITOR

The Ownship Navigation Monitor shall be identical to Process 2.3, except that the timing constraints are less rigid.

This module shall interface with the ownship navigation system. The ownship latitude, longitude, course, velocity and current time are needed for reporting purposes. While this process must operate under real-time constraints, they are less stringent than those of Process 2.3.

Provided that "ownship" is, in fact, a mobile platform, and is equipped with a navigation system (or set of navigation systems), this module is the workstation interface to the navigation system(s). This module would be concerned with the message protocols and bit patterns necessary to retrieve ownship position information from the navigation system(s), and translate that information into a standardized format.

The Ownship Navigation Monitor module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

> A navigation system information update (*binary data stream*) is received and a significant update of ownship latitude, longitude, course, velocity or time has occurred.
> [The frequency of positional information update will be a function of "ownship" speed. The faster the motion, the more frequently the information should be updated.]

Postcondition

> When the precondition is met, this module shall forward *ownship navigation information* to Ownship Track Generator (3.4.2).

### 3.4.2 OWNSHIP TRACK GENERATOR

Ownship navigation information will be periodically updated. The Ownship Track Generator process shall put this information into a special ownship track.

The Ownship Track Generator module shall execute periodically at a one second interval (for ships; aircraft may require a faster interval). This module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

> *Ownship navigation information* is received and execution interval expires.

Postcondition

> When the precondition is met, this module shall forward a *update track tuple* request for ownship track to Track Database Update (3.1).

# 4 TRACK CONTROLLER

## 4.1 TRACK MANAGER DIALOGUE

### 4.1.1 INITIALIZE CONSTRAINTS

#### 4.1.1.1 CONSTRAINT SELECTION

The Constraint Selection process shall provide the user with the ability to reset any user specified variable relating to track database functions. The user may choose a general category of variables to verify or alter.

The Constraint Selection module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

A *terminal input* is received, indicating user's desire to alter track management constraints.

Postcondition

When the precondition is met, this module shall display categories of track management constraints that the user may alter.
[Currently, the user may alter the following:
*initialize transmission sequence options*,
*archive setup options*,
*[illegible] constraints options*, and
*track filter options*.]

#### 4.1.1.2 TRANSMISSION SEQUENCE MENU

The Transmission Sequence Menu process shall provide the user a comprehensive and interactive means for initializing a periodic communications transmission.

The Transmission Sequence Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

The *initialize transmission option* is selected.

Precondition 2

*Transmission sequence data* are received.

Precondition 3

Current values are approved, and change in system values are approved.

Precondition 4

Cancel transmission sequence initialization is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify default values.

Postcondition 3

When precondition 3 is met, this module shall forward the *initiate transmission sequence* command to the Periodic Transmission Generator (1.6).

Postcondition 4

When precondition 4 is met, this module shall do nothing.

### 4.1.1.3 ARCHIVE SETUP MENU

The Archive Setup Menu process shall provide the user a comprehensive and interactive means for delineating the type of messages to be saved and processed by the C3I workstation.

The Archive Setup Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

The *archive setup option* is selected.

Precondition 2

*Archive setup data* are received.

Precondition 3

Current values are approved, and change in system values are approved.

Precondition 4

Cancel archive setup initialization is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify default values.

Postcondition 3

When precondition 3 is met, this module shall forward *archive setup* command to the Message Librarian (1.2).

Postcondition 4

When precondition 4 is met, this module shall do nothing.

#### 4.1.1.4 TRACK MONITOR MENU

The Track Monitor Menu process shall provide the user a comprehensive and interactive means for delineating variables that control the behavior of the track Database Monitor (3.3).

The Track Monitor Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

The *monitor constraints option* is selected.

Precondition 2

*Track monitor data* are received.

Precondition 3

Current values are approved, and change in system values are approved.

Precondition 4

Cancel track monitor initialization is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify default values.

Postcondition 3

When precondition 3 is met, this module shall forward *set monitor constraints* command to the Database Monitor (3.3).

Postcondition 4

When precondition 4 is met, this module shall do nothing

### 4.1.1.5 TRACK FILTER MENU

The Track Filter Menu process shall provide the user a comprehensive and interactive means for delineating the type of tracks to be maintained by the track database.

The Track Filter Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

The *track filter option* is selected.

Precondition 2

*Track filter data* are received.

Precondition 3

Current values are approved, and change in system values are approved.

Precondition 4

Cancel track filter initialization is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify Default values.

Postcondition 3

When precondition 3 is met, this module shall forward *set track filter* command to the Track Update (3.1).

Postcondition 4

When precondition 3 is met, this module shall do nothing.

## 4.1.2 DATABASE MANIPULATION

### 4.1.2.1 DATABASE FUNCTION SELECTION

The Database Function Selection process shall provide the user with a choice of database related functions. The user may choose a general category of actions of interest.

The Database Function Selection module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Terminal input* is selected, indicating user's desire to perform a database related function.

Postcondition

When the precondition is met, this module shall display function list. [Currently, the user may perform any of the following:
(textually) *display tracks option* data,
*add track option* to the database,
*update track option* within the database, and
*delete track option* from the database.]

### 4.1.2.2 TRACK DISPLAY MENU

The Track Display Menu process shall provide the user a comprehensive and interactive means for querying the track database. For the sake of example, if the user were interested in seeing a textual representation of the information associated with track # T002344, then it should only be a matter of selecting a track ID search on "T002344".

The database query language shall provide robust, yet simple data queries, such as "all air tracks within 200 nautical miles." A sophisticated language interface shall permit the user (user) some flexibility of format, as well as meaningful error messages.

The Track Display Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

A *display tracks option* command is received.

Precondition 2

A *database query* is received.

Precondition 3

Cancel database query action is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu.

Postcondition 2

When precondition 2 is met, this module shall forward *display tracks call* to Retrieve Track Tuple Information (4.2).

Postcondition 3

When precondition 3 is met, this module shall do nothing.

### 4.1.2.3 ADD TRACK MENU

The Add Track Menu process shall provide the user a comprehensive and interactive means for adding a new track tuple into the track database.

The Add Track Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

An *add track option* command is received.

Precondition 2

An *add track data* is received.

Precondition 3

Cancel add track action is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu.

Postcondition 2

When precondition 2 is met, this module shall forward *track tuple* to Add New Track to Database (4.3).

Postcondition 3

When precondition 3 is met, this module shall do nothing.

#### 4.1.2.4 UPDATE TRACK MENU

The Update Track Menu process shall provide the user a comprehensive and interactive means for modifying existing track tuple information within the track database.

The Update Track Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition 1

An *update track option* command is received.

Precondition 2

*Update track data* are received.

Precondition 3

Cancel update track action is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu.

Postcondition 2

When precondition 2 is met, this module shall forward *track tuple* to Modify Existing Track Data (4.4).

Postcondition 3

When precondition 3 is met, this module shall do nothing.

### 4.1.2.5 DELETE TRACK MENU

The Delete Track Menu shall provide the user a comprehensive and interactive means for removing a track tuple from the track database.

The Delete Track Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated whe. .ny of the following preconditions occurs:

Precondition 1

A *delete track option* command is received.

Precondition 2

*Delete track data* are received.

Precondition 3

Cancel delete track action is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu.

Postcondition 2

When precondition 2 is met, this module shall forward *track ID* to Delete Track From Database (4.5).

Postcondition 3

When precondition 3 is met, this module shall do nothing.

## 4.2 RETRIEVE TRACK TUPLE INFORMATION

The Retrieve Track Tuple Information process shall provide non-graphical track information support to the user (human operator). If the user is interested in comparing track attribute values between two or more surface contacts, then this module shall interface with the Track Database Manager (3) to retrieve the desired information.

The Retrieve Track Tuple module shall have a maximum execution time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Display tracks call* is received.

Postcondition

When the precondition is met, this module shall generate a Track *database request* and shall send it to Track Request (3.2). This module shall also retrieve *track tuple* information, and forward the information to the Track Manager Display (4.6).

## 4.3 ADD NEW TRACK TO DATABASE

The Add New Track To Database process shall submit a track tuple to the Track Database Manager (3) for inclusion into the track database.

The Add New Track To Database module shall have a maximum execution time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

A *track tuple* is received.

Postcondition

When the precondition is met, this module shall send an *add track tuple* request to Track Database Update (3.1).

## 4.4 MODIFY EXISTING TRACK DATA

The Modify Existing Track Data process shall submit a track tuple update request to the Track Database Manager (3).

The Modify Existing Track Data module shall have a maximum execution time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

A *track tuple* is received.

Postcondition

When the precondition is met, this module shall send an *update track tuple* request to Track Database Update (3.1).

## 4.5 DELETE TRACK FROM DATABASE

The Delete Track From Database process shall submit a delete track tuple request to the Track Database Manager (3).

The Delete Track From Database module shall have a maximum execution time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

A *track ID* is received.

Postcondition

When the precondition is met, this module shall send a *delete track tuple* request to Track Database Update (3.1).

## 4.6 TRACK MANAGER DISPLAY

### 4.6.1 DISPLAY RESOLUTION SCREEN

The resolution notice shall be produced by the Database Monitor (3.3) whenever it determines that two or more tracks share enough information in common such that there is a likelihood that these tracks are actually one track redundantly entered into the track database.

The Display Resolution Screen process shall produce a pop-up window that clearly explains to the user (human operator) the nature of the track ambiguity resolution notice. The user is left to decide the appropriate course of action to take -- whether to delete one or more track, or to match merge and correlate the given information.

The Display Resolution Screen module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition

A *resolution notice* is received.

Postcondition

When the precondition is met, this module shall display the Track ambiguity *resolution screen*.

### 4.6.2 DISPLAY INTEL REPORT SCREEN

The intelligence report shall be produced by the Sensor Interface Control (2.1) whenever a sensor produces information that does not clearly correspond to track tuple attributes, and yet is clearly of tactical importance.

The Display Intel Report Screen process may contain an expert system of a Tactical Action Officer (TAO) that is capable of correlating intelligence information and deducing tactical information. (E.g., if an ESM device were to report that it was intercepting electromagnetic emissions at a particular spectral bandwidth, and it was known that the only known radar system that operates at that particular frequency was the Soviet Downbeat radar, and that the Downbeat radar is only installed on Soviet Bear aircraft (Tu-142), then the expert system could conclude (and advise) that a probable Bear aircraft was in the vicinity.)

This module shall produce a pop-up window that clearly presents to the user (human operator) the given intelligence report (and optional conclusions). The user is left to decide the appropriate course of action to take -- whether to add one or more tracks, or to associate the given information with an existing track (e.g., if a Tu-142 aircraft is already in the vicinity or direction of the report, then the intelligence information could be associated with that track).

The Display Intel Report Screen module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition

An *intelligence report* is received.

Postcondition

When the precondition is met, this module shall display an *Intelligence report screen*.

### 4.6.3 DISPLAY TRACK TUPLE SCREEN

The Display Track Tuple Screen process shall produce a pop-up window that clearly presents to the user (human operator) the set of track information that satisfies the original display tracks call. With this screen displayed, the user shall be able to update or delete track information.

The Display Track Tuple Screen module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Track tuples* are received from Retrieve Track Tuple Information (4.2).

Postcondition

When the precondition is met, this module shall display the *Track tuples screen*.
[Track information shall be formatted and clearly displayed.]

# 5 TACTICAL COMMAND DISPLAY

## 5.1 BATTLE MANAGER DIALOGUE

### 5.1.1 SYSTEM INPUT

#### 5.1.1.1 FUNCTION SELECTION

##### 5.1.1.1.1 BATTLE MANAGER FUNCTION SELECTION

The Battle Manager Function Selection process shall provide the user with a choice of information access and dissemination functions. The user may choose a general category of actions of interest.

The Battle Manager Function Selection module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Terminal input* is selected, indicating the user's desire to perform a battle management related function.

Postcondition

When the precondition is met, this module shall display the function list.
[Currently, the user may perform any of the following:
display ownship platform status,
(graphically) display track data,
generate/edit a communications message, and
display (view) a communications message.]

##### 5.1.1.1.2 STATUS MENU

The Status Menu process shall provide the user a comprehensive and interactive means for determining the current status of ownship or a particular weapon system aboard ownship.

The Status Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

*Display status option* command is received.

Precondition 2

*Status input* data are received.

Precondition 3

Current values are approved.

Precondition 4

Cancel status query is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify default values.

Postcondition 3

When precondition 3 is met, this module shall forward a *platform status call* to the Platform Status Monitor (5.2).

Postcondition 4

When precondition 4 is met, this module shall do nothing.

### 5.1.1.1.3 TRACK PLOT MENU

The Track Plot Menu process shall provide the user a comprehensive and interactive means for delineating a graphical track display window. The user shall be prompted to specify the size of the display window, geographic region to be display within the window, and the window display refresh rate (which may be very different from the track database refresh rate).

The Track Plot Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

*Display track option* command is received.

Precondition 2

*Track display data* are received.

Precondition 3

Current values are approved.

Precondition 4

Cancel track display is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify default values.

Postcondition 3

When precondition 3 is met, this module shall forward *track display call* to Graphical Track Display (5.2).

Postcondition 4

When the precondition 4 is met, this module shall do nothing.

#### 5.1.1.1.4 GENERATE MESSAGE MENU

The Generate Message Menu process shall provide the user a comprehensive and interactive means for generating or editing a communications message, including header and routing information.

The Generate Message Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

*Generate message option* command is received.

Precondition 2

*Edit data* are received.

Precondition 3

Current values are approved.

Precondition 4

Cancel editing session is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify default values.

Postcondition 3

When precondition 3 is met, this module shall forward *edit commands* to the Message Generator (5.4).

Postcondition 4

When precondition 4 is met, this module shall do nothing.

#### 5.1.1.1.5 VIEW MESSAGE MENU

The View Message Menu process shall provide the user a comprehensive and interactive means for locating an extant textual message resident within the system memory.

The View Message Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

*View message option* command is received.

Precondition 2

*Read message data* are received.

Precondition 3

Current values are approved.

Precondition 4

Cancel read message is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify default values .

Postcondition 3

When precondition 3 is met, this module shall forward *read message call* to the Textual Message Display (5.6).

Postcondition 4

When precondition 4 is met, this module shall do nothing.

### 5.1.1.2 NETWORK CONSTRAINT SELECTION

#### 5.1.1.2.1 NETWORK COMMAND OPTIONS

The Network Command Options process shall provide the user with the ability to reset any user specified variable relating to network communications functions. The user may choose a general category of variables to verify or alter.

The Network Command Options module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

Terminal input is selected, indicating user's desire to alter communications network constraints.

Postcondition

When the precondition is met, this module shall display categories of network constraints that the user may alter.
[Currently, the user may alter the following:
network setup options, and
emissions status options.]

#### 5.1.1.2.2 STATUS REPORT MENU

The Status Report Menu process shall provide the user a comprehensive and interactive means for initializing standardized information gathering and reporting constraints.

The Status Report Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

*Set reporting option* is selected.

Precondition 2

*Reporting data* are received.

Precondition 3

Current values are approved, and change in system values are approved.

Precondition 4

Cancel reporting initialization is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify default values.

Postcondition 3

When precondition 3 is met, this module shall forward *reporting setup* command to the Status Report Generator (5.7).

Postcondition 4

When precondition 4 is met, this module shall do nothing.

#### 5.1.1.2.3 NETWORK SETUP MENU

The Network Setup Menu process shall provide the user a comprehensive and interactive means for initializing communications network constraints.

The Network Setup Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

*Network setup option* is selected.

Precondition 2

*Network data* are received.

Precondition 3

Current values are approved, and change in system values are approved.

Precondition 4

Cancel network constraint initialization is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify default values.

Postcondition 3

When precondition 3 is met, this module shall forward *network setup* command to the Communications Network Monitor & Control (1.4).

Postcondition 4

When precondition 4 is met, this module shall do nothing.

### 5.1.1.2.4 EMISSIONS STATUS MENU

The Emissions Status Menu process shall provide the user a comprehensive and interactive means for setting communications transmission constraints.

The Emissions Status Menu module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

*Emissions setup option* is selected.

Precondition 2

*Emissions data* are received.

Precondition 3

Current values are approved, and change in system values are approved.

Precondition 4

Cancel emission constraint initialization is selected.

Postcondition 1

When precondition 1 is met, this module shall display a pop-up menu, indicating default values.

Postcondition 2

When precondition 2 is met, this module shall modify default values.

Postcondition 3

When precondition 3 is met, this module shall forward *emissions control command* to the Communications Network Monitor & Control (1.4).

Postcondition 4

When precondition 4 is met, this module shall do nothing.

### 5.1.2 SYSTEM OUTPUT

#### 5.1.2.1 DISPLAY EMERGENCY STATUS REPORT

When a weapon system is no longer functioning, whether due to damage, failure or maintenance, the user shall be notified. The Display Emergency Status Report process shall display a pop-up screen or command line with explanatory text.

Depending on the importance of this consideration, a particular implementation may choose to require the user to acknowledge having read the screen (or command line) prior to the screen being refreshed.

The Display Emergency Status Report module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition

An *emergency weapon status report* is received.

Postcondition

When the precondition is met, this module shall display an *emergency status screen.*

### 5.1.2.2 MESSAGE ARRIVAL DISPLAY

When a new communications message or LAN electronic mail message is received, the user shall be notified. The Message Arrival Display process shall display a pop-up screen or command line identifying message header information (To whom, From whom, Subject, etc.).

Depending on the importance of this consideration, a particular implementation may choose to require the user to acknowledge having read the screen (or command line) prior to the screen being refreshed.

The Message Arrival Display module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*New message notice* is received.

Postcondition

When the precondition is met, this module shall display a *message arrival notification.*

### 5.1.2.3 DISPLAY WEAPON STATUS

In response to the user's weapon system *status query*, the Display Weapon Status process shall display a pop-up window which includes the requested information.

The Display Weapon Status module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Status report* is received.

Postcondition

When the precondition is met, this module shall display a *weapon status screen.*

### 5.1.2.4 TRACK DISPLAY

In response to the user's *track display call,* the Track Display process shall display a pop-up window that shall graphically display and periodically update the requested track information.

The Track Display module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Graphics display* is received.

Postcondition

When the precondition is met, this module shall display the *graphics display screen.*

### 5.1.2.5 DISPLAY EDIT SCREEN

In response to the user's *edit command,* the Display Edit Screen process shall display a pop-up window and interactive dialogue for creating naval messages. For constructing a message header, the edit dialogue shall have templates for entering the following information that is common to all Navy messages, and shall also have separate templates for specified formatted messages (e.g., OPREP, SITREP, JINTACCS, etc.).

The Display Edit Screen module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Edit prompt* is received.

Postcondition

When the precondition is met, this module shall display the appropriate *edit screen.*

### 5.1.2.6 DISPLAY TEXT FILE

In response to the user's *read message call*, the Display Text File process shall display a pop-up window with the designated textual message.

Under certain circumstances it may be desirable to incorporate the text from an existing message within another message. Hence, a mechanism whereby certain information may be stored within a local buffer may be desirable.

The Display Text File module shall have a maximum execution time of 200 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Text file* is received.

Postcondition

When the precondition is met, this module shall display the *text screen*.
[The user shall have the opportunity to scroll through the given text at his leisure.]

## 5.2 PLATFORM STATUS MONITOR

The Platform Status Monitor shall be a periodic process which shall maintain weapon system statuses. The status information shall be updated on a semi-regular basis (once every sixty seconds), or upon request.

The Platform Status Monitor module shall execute periodically (user defined interval) and shall have a maximum execution time of 10 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

Periodic time interval update.

Precondition 2

*Platform status call* is received from Function Selection (5.1.1.1), that requests general platform status information.

Precondition 3

*Platform status call* is received from Function Selection (5.1.1.1), that requests specific platform status information.

Precondition 4

*Status query* is received from Status Report Generator (5.7).

Postcondition 1

When precondition 1 is met, this module shall forward the *Status query* to Ownship Weapon Status Monitor (6.1) in reference to all weapons systems resident on ownship platform.

Postcondition 2

When precondition 2 is met, this module shall forward the *Status report* immediately to System Output (5.1.2) based upon most recent information.

Postcondition 3

When precondition 3 is met, this module shall forward the *Status query* to Ownship Weapon Status Monitor (6.1) in reference to a specific weapon system Resulting *status report* shall be forwarded to System Output (5.1.2).

Postcondition 4

When precondition 4 is met, this module shall forward the *Status report* immediately to Status Report Generator (5.7) based upon most recent information.

## 5.3 GRAPHICAL TRACK DISPLAY

### 5.3.1 TRACK DISPLAY MONITOR

Since the C3I workstation will support a multi-windowing environment, a graphical track display shall be an independent window within the system. Further the C3I workstation shall permit multiple graphical track displays to be active simultaneously.

For example, the user shall be capable of viewing two or more separate windows at the same time (e.g., one with air tracks within 500 nautical miles, another with surface tracks within 300 nautical miles, and yet another with subsurface tracks within 50 nautical miles).

The Track Display Monitor process shall provide real-time multi-tasking multi-window display capability. This module shall periodically choreograph the track display process by creating a track display window, maintaining geographic references with corresponding map overlays and geometric frames of reference, as well as providing periodic track updates.

The Track Display Monitor module have a maximum execution time of 20 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Track display call* is received.
[E.g., "Display all air tracks within 500 nmi of (ownship) x°N latitude, y°W longitude." or "Display all reported tracks within (the Black Sea) South of xx°N and North of yy°N, East of pp°E and West of qq°E."]

Postcondition

When the precondition is met, this module shall:
Issue a *map request* to Map Generator (5.3.2).
Issue a *window request* to Window Generator (5.3.3).
Issue a *track request* to Track Display Generator (5.3.4).
Issue a *geometric display request* to Geometric Display Generator (5.3.5).

This module shall also combine resulting *map*, *graphics tracks* and *geometric displays* within the track display *window*.in order to produce the net *graphics display* and forward the *graphics display* to System Output (5.1.2).

Finally, this module shall update this information periodically.

### 5.3.2 MAP GENERATOR

The Defense Mapping Agency (DMA) provides digitized map databases for use by the Department of Defense. The Map Generator process shall be specifically designed to read DMA map data (e.g., World Database II) and produce system specific graphical data for use in the given windowing environment. (See Figure D-4.)

The Map Generator module shall have a maximum response time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Map request* is received.

Postcondition

When the precondition is met, this module shall parse through DMA map database, extract that portion relevant to the given *map request* constraints (i.e., within specified latitude and longitude boundaries), and convert the DMA map format into a system specific graphical format.

This module shall also forward system displayable *map* to Track Display Monitor (5.3.1).

Figure D-4. A map of Europe

### 5.3.3 WINDOW GENERATOR

The Window Generator process shall interface with software resident on the machine (hosting the C3I workstation) that provides for a multi-windowing environment.

The Window Generator module shall have a maximum response time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Window request* is received.

Postcondition

When the precondition is met, this module shall create a new window and forward the *window* parameters to Track Display Monitor (5.3.1).

### 5.3.4 TRACK DISPLAY GENERATOR

The Track Display Generator process shall retrieve track information for the purpose of displaying NTDS (or NTDS follow-on) symbols on the graphical track display. [Note: not all track information stored within a track tuple is necessary for display purposes.]

The Track Display Generator module shall execute periodically (at a user defined interval) and shall have a maximum response time of 1000 ms due to Track Request (3.2) processing limitations. If Track Request s can be returned more quickly, then the response time of this module may be improved.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

*Track request* is received.

Precondition 2

*Track tuple* information is received.

Postcondition 1

When precondition 1 is met, this module shall forward the *database request* to Track Database Manager (3).

Postcondition 2

When precondition 2 is met, this module shall filter out unnecessary track information and forward *graphics tracks* to Track Display Monitor (5.3.1).

### 5.3.5 GEOMETRIC DISPLAY GENERATOR

For the purposes of establishing frames of reference, the user may wish to display map related geometric references within a graphical track display.

For example, if a capital ship is designated as the "force center of operations", then a circular grid reference may be displayed which may be centered over the capital ship's coordinates. Also, during operations against Lybia a few years ago, Ghadaffi's "Line of Death" would have been helpful to display.

The Geometric Display Generator process shall generate the appropriate geometric figure(s), drawn to scale, for the purposes of overlaying onto the track and map graphical display.

The Geometric Display Generator module shall have a maximum response time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Geometric display request* is received.

Postcondition

When the precondition is met, this module shall forward the prescribed *geometric display* to Track Display Monitor (5.3.1).

## 5.4 MESSAGE GENERATOR

### 5.4.1 EDIT DIALOGUE

#### 5.4.1.1 MESSAGE SELECTION

The Message Selection process shall provide the user with a choice of information access and dissemination functions. The user may choose a general category of actions of interest.

The Message Selection module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Edit command* is selected, indicating user's desire to perform a text editing function.

Postcondition

When the precondition is met, this module shall display the function list. [Currently, the user may perform any of the following:
create a standard format message,
create an unformatted text message, or
edit an existing text file.]

### 5.4.1.2 RETRIEVE TEMPLATE

The Retrieve Template process shall act as an interface with the Create New File (5.4.2) process.

The Retrieve Template module shall have a maximum execution time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Format request* is received.

Postcondition

When the precondition is met, this module shall send a *format request* to Create New File (5.4.2). The resulting *text file* (message template) shall be forwarded to the Text Editor (5.4.1.4).

### 5.4.1.3 RETRIEVE EXISTING FILE

The Retrieve Existing File process shall act as an interface with the Read Existing File (5.4.3) process.

The Retrieve Existing File module shall have a maximum execution time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*File request* is received.

Postcondition

When the precondition is met, this module shall send a *file request* to Read Existing File (5.4.3). The resulting *text file* shall be forwarded to the Text Editor (5.4.1.4).

### 5.4.1.4 TEXT EDITOR

The Text Editor process shall provide standard keyboard entry of text, and textual support functions such as a cut-and-paste capability to enable the user to compose the body of a.message, periodic auto-save, etc.. A line counter and a column indicator shall be provided to allow feedback on message length and to provide information required for precise formatting. Cursor positioning shall be performed using an attached cursor positioning devices (e.g., trackball or mouse) with selection buttons to support a modern text editing interface for selecting, moving and deleting of text.

The Text Editor module shall have a maximum execution time of 200 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

Null *text file* is received from Message Selection (5.4.1.1).

Precondition 2

Message template *text file* is received from Retrieve Template (5.4.1.2).

Precondition 3

*Text file* is received from Retrieve Existing File (5.4.1.3).

Precondition 4

Include track data command is selected.
A pop-up database request menu is displayed to accept *database request.*
Track *database request is* received.

Precondition 5

Save text command is selected and *filename* is provided.

Precondition 6

Transmit message command is selected.

Precondition 7

Keyboard data is entered.

Precondition 8

Edit function is selected.

Postcondition 1

When precondition 1 is met, this module shall select the text edit mode , and provide a clean slate (empty buffer) for keyboard data entry.

Postcondition 2

When precondition 2 is met, this module shall select text edit mode, and display the *text file* for modification.

Postcondition 3

When precondition 3 is met, this module shall select the text edit mode, and display the *text file* for modification.

Postcondition 4

When precondition 4 is met, this module shall forward the database request to Receive Track Text Data (5.4.1.5). The returned textual track tuple information (*text file*) shall be inserted into the message edit buffer at the cursor location.

Postcondition 5

When precondition 5 is met, this module shall forward the *save command* to Save Text File (5.4.1.7).

Postcondition 6

When precondition 6 is met, this module shall forward the contents of message edit buffer (*text file*) to Send Text File (5.4.1.6).

Postcondition 7

When precondition 7 is met, this module shall insert keyboard data into message edit buffer at the current cursor position.

Postcondition 8

When precondition 7 is met, this module shall perform the appropriate/designated edit function upon the message edit buffer.

### 5.4.1.5 RETRIEVE TRACK TEXT DATA

The Retrieve Track Text Data process shall serve as another means for including textual information concerning tracks within a message. Whereas the Status Report Generator (5.7) shall include all track data that satisfies the report defaults, this process shall enable the user to provide a tailored subset of track information within a message.

The Retrieve Track Text Data module shall have a maximum execution time of 100 ms

This module shall be initiated when the following precondition occurs:

Precondition

*Database request* is received.

Postcondition

When the precondition is met, this module shall send the *database request* to the Track Database Manager (3). The resulting *track tuple* information shall be placed into textual format. This module shall format *track tuple* information and forward it to the Text Editor (5.4.1.4).

### 5.4.1.6 SEND TEXT FILE

Once a message has been generated, the Send Text File process shall forward the message for transmission. This module shall also perform a completion check in order to verify that all requisite message information is filled in properly.

The Send Text File module shall have a maximum execution time of 100 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

*Text file* is received, with valid message format.

Precondition 2

*Text file* is received, with invalid message formatting.

Postcondition 1

When precondition 1 is met, this module shall send the *text file* to Message Processor (5.5).

Postcondition 2

When precondition 2 is met, this module shall highlight *format error*, and prompt the user for correction.

### 5.4.1.7 SAVE TEXT FILE

When the user specifies a message to be saved, the Save Text File process shall write a copy to memory.

The Save Text File module shall have a maximum execution time of 100 ms.

This module shall be initiated when the following precondition occurs:

Precondition

*Save command* is received.

Postcondition

When the precondition is met, this module shall store the *text file* in the Miscellaneous Text Files database, under the specified *filename*.

### 5.4.2 CREATE NEW FILE

The C3I workstation shall provide a text message standard format *template archive*, wherein header and message body format information is stored. The Create New File process shall process message *format requests*, access the template archive database, and return a text file that contains the appropriate message template.

Further, for messages that require pre-specified data elements (e.g., a SITREP or status report), this module shall select the appropriate message template and automatically fill in the appropriate data fields before returning the template to the text editor.

The Create New File module shall have a maximum execution time of 100 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

*Format request* is received, and the template has no pre-specified data elements.

Precondition 2

*Format request* is received, and the template contains pre-specified data elements.

Postcondition 1

When precondition 1 is met, this module shall retrieve the standard message format template from the Template Archive database. This template shall be forwarded to the Edit Dialogue (5.4.1), as an ASCII *text file*.

Postcondition 2

When precondition 2 is met, this module shall retrieve the standard message format template from the Template Archive database. Pre-specified data elements shall be identified and forwarded to the Status Report Generator (5.7). The resulting data elements shall be returned, and placed into appropriate locations within the message template file. The expanded template (template with data elements filled in) shall be forwarded to the Edit Dialogue (5.4.1), as an ASCII *text file*.

### 5.4.3 READ EXISTING FILE

The Read Existing File process shall locate and retrieve text files for the purpose of editing them. *Text files* may already reside as ASCII files stored in system memory, or they may be text messages currently being displayed for the user. Hence, instead of creating a message "from scratch" this process supports text reuse.

The Read Existing File module shall have a maximum execution time of 100 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

*File request* is received, and the requested file already stored in Miscellaneous Text Files.

Precondition 2

*File request* is received, and the requested file currently being displayed.

Postcondition 1

When precondition 1 is met, this module shall forward the prescribed *text file* to Edit Dialogue (5.4.1).

Postcondition 2

When precondition 2 is met, this module shall forward the prescribed message (*text file*) to Edit Dialogue (5.4.1).

## 5.5 MESSAGE PROCESSOR

The Message Processor shall primarily function as a queue for inbound and outbound messages. Auxiliary functions for this module may include queuing messages according to precedence, verifying the presence (and possibly syntax) of key message elements,.and checking the message security level (to insure that classified messages are not sent over networks with with lower classification).

The Message Processor module shall have a maximum execution time of 100 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

Inbound message is received.

Precondition 2

Outbound message is received.

Postcondition 1

When precondition 1 is met, this module shall forward the *text file* to Textual Message Display (5.6)

Postcondition 2

When precondition 2 is met, this module shall forward the outgoing message to Communications Network Monitor & Control (1.4), via *electronic mail.*

## 5.6 TEXTUAL MESSAGE DISPLAY

### 5.6.1 MESSAGE RETRIEVAL

The Message Retrieval process shall respond to read message calls, initiated by the user. The goal of this process shall be to locate the specified text file and pass it back to the Battle Manager Dialogue (5.1).

The Message Retrieval module shall have a maximum response time of 100 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

*Read message call* is received, and the specified text file resides in the Incoming Message Queue (5.6.2).

Precondition 2

*Read message call* is received, and the specified text file resides in the Message Archives.

Precondition 3

*Read message call* is received, and the specified text file resides in the Miscellaneous Text Files.

Postcondition 1

When precondition 1 is met, this module shall send a *message request* to Incoming Message Queue (5.6.2) and forward the returned *text file* to System Output (5.1.2).

Postcondition 2

When precondition 2 is met, this module shall read the designated *text file* from the Message Archives and forward the designated *text file* to System Output (5.1.2).

Postcondition 3

When precondition 3 is met, this module shall read the designated *text file* from the Miscellaneous Text Files and forward the designated *text file* to System Output (5.1.2).

### 5.6.2 INCOMING MESSAGE QUEUE

Messages to be read by the user, whether they originate from somewhere within ownship (via platform local area network electronic mail) or from some other U.S. Naval platform or land base (via tactical communications), shall be queued and stored until such time that they may be read and acted upon by the Incoming Message Queue process.

Whenever a message is placed into the queue, a new message notice shall alert the user to the newly arrived message.

The Incoming Message Queue module shall have a maximum execution time of 100 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

*Text file* is received from Message Processor (5.5).

Precondition 2

*Read message request* is received from Message Retrieval (5.6.1).

Postcondition 1

When precondition 1 is met, this module shall queue the *text file* and store it for later reference. *New message notice* shall be sent to System Output (5.1.2).

Postcondition 2

When precondition 2 is met, this module shall remove the designated *text file* from the queue, and forward it to System Output (5.1.2).

## 5.7 STATUS REPORT GENERATOR

The Status Report Generator process shall support the Message Generator (5.4), by providing an automatic means for retrieving track information and ownship status information for later inclusion into the textual body of a communications message.

The Status Report Generator module shall have a maximum execution time of 1000 ms.

This module shall be initiated when any of the following preconditions occurs:

Precondition 1

*Reporting setup* command is received.

Precondition 2

*Status report call* is received, requesting ownship status information.

Precondition 3

*Status report call* is received, requesting track report information.

Precondition 4

*Status report call* is received, requesting both ownship status information as well as track report information.

Postcondition 1

When precondition 1 is met, this module shall set local variables accordingly.
[These variables shall provide information specific to a particular reporting format. For example, if an OTCIXS SITREP information is requested, then local default variables would be verified to determine what type of track information is to be returned (i.e., all known air, surface and subsurface contacts within 50 nautical miles).]

Postcondition 2

When precondition 2 is met, this module shall forward the ownship *status query* to Platform Status Monitor (5.2). The resulting *status report* shall be converted into textual format and forwarded to Message Generator (5.4).

Postcondition 3

When precondition 3 is met, this module shall forward the track *database request* to Track Database Manager (3). The resulting *track tuple* information shall be converted into textual format and forwarded to Message Generator (5.4).

Postcondition 4

When precondition 4 is met, this module shall forward the ownship *status query* to Platform Status Monitor (5.2). This module shall also forward the track *database request* to Track Database Manager (3). The resulting *status report* and *track tuple* information shall be converted into textual format and forwarded to Message Generator (5.4).

# 6 WEAPONS SYSTEMS INTERFACE

## 6.1 OWNSHIP WEAPONS STATUS MONITOR

Based upon the nature of the *status query*, the Ownship Weapons Status Monitor process shall in turn query a given set of weapon systems concerning their operational status, loadout, etc..

The Ownship Weapons Status Monitor module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

A weapon system *status query* is received.

Postcondition

When the precondition is met, this module shall forward a weapon system *status report* Platform Status Monitor (5.2).

## 6.2 EMERGENCY STATUS REPORTER

If any event has transpired that would make a weapon system inoperable (damage, failure or maintenance) then the Emergency Status Reporter process shall notify the user of this action.

The Emergency Status Reporter module shall have a maximum execution time of 10 ms.

This module shall be initiated when the following precondition occurs:

Precondition

An *emergency weapon status report* is rceeived.

Postcondition

When the precondition is met, this module shall forward an *emergency weapon status report* to Battle Manager Dialogue (5.1).

## 6.3 CIWS STATUS CONTROL

Provided that ownship is equipped with a PHALANX close-in weapon system (CIWS), then the CIWS Status Control process shall be the workstation interface to that system. This module shall be concerned with the message protocols and bit patterns necessary to monitor the weapon system status.

The CIWS Status Control module shall have a maximum execution time of 1000 ms and a maximum response time of 500 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A weapon system *status update request* is received.

Precondition 2

A change in weapon system status is detected (or received).

Postcondition 1

When precondition 1 is met, this module shall forward a weapon system *status report* to Ownship Weapons Status Monitor (6.1).

Postcondition 2

When precondition 2 is met, this module shall forward an *emergency weapon status report* to Emergency Status Reporter (6.2).

## 6.4 5"/54 GUN WEAPON SYSTEM STATUS CONTROL

Provided that ownship is equipped with a 5"/54 gun weapon system, with a Mk 86 digital gun fire control system (GFCS), then the 5"/54 Gun Weapon System Status Control process shall be the workstation interface to that system. This module shall be concerned with the message protocols and bit patterns necessary to monitor the weapon system status.

The 5"/54 Gun Weapon System Status Control module shall have a maximum execution time of 1000 ms and a maximum response time of 500 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A weapon system *status update request* is received.

Precondition 2

A change in weapon system status is detected (or received).

Postcondition 1

When precondition 1 is met, this module shall forward a weapon system *status report* to Ownship Weapons Status Monitor (6.1).

Postcondition 2

When precondition 2 is met, this module shall forward an *emergency weapon status report* to Emergency Status Reporter (6.2).

## 6.5 TWS STATUS CONTROL

Provided that ownship is equipped with the Tomahawk weapon system, then the TWS Status Control process shall be the workstation interface to the Tomahawk weapon control system (TWCS). This module shall be concerned with the message protocols and bit patterns necessary to monitor the weapon system status.

The TWS Status Control module shall have a maximum execution time of 1000 ms and a maximum response time of 500 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A weapon system *status update request* is received.

Precondition 2

A change in weapon system status is detected (or received).

Postcondition 1

When precondition 1 is met, this module shall forward a weapon system *status report* to Ownship Weapons Status Monitor (6.1).

Postcondition 2

When precondition 2 is met, this module shall forward an *emergency weapon status report* to Emergency Status Reporter (6.2).

## 6.6 MK46 TORPEDO STATUS CONTROL

Provided that ownship is equipped with a modern surface launched Mark 46 torpedo delivery system, then the MK46 Torpedo Status Control process shall be the workstation interface to the Mk 116 underwater fire control system (UFCS). This module shall be concerned with the message protocols and bit patterns necessary to monitor the torpedo weapon system status.

The MK46 Torpedo Status Control module shall have a maximum execution time of 1000 ms and a maximum response time of 500 ms.

This module shall be initiated when either of the following preconditions occurs:

Precondition 1

A weapon system *status update request* is received.

Precondition 2

A change in weapon system status is detected (or received).

Postcondition 1

When precondition 1 is met, this module shall forward a weapon system *status report* to Ownship Weapons Status Monitor (6.1).

Postcondition 2

When precondition 2 is met, this module shall forward an *Emergency weapon status report* to Emergency Status Reporter (6.2).

# APPENDIX E

## DATA DICTIONARY

| | | |
|---|---|---|
| **addressee** | = { string } | |
| | * the intended recipient of a given message | * |
| **add_track_call** | = track_tuple | |
| **add_track_tuple** | = origin + ADD + track_tuple | |
| **alias** | = { string } | |
| | * this provides for easier selection of commonly addressed | * |
| | * messages from a user built library. For example, by using | * |
| | * aliases, the user may enter the word "squadron" as the | * |
| | * addressee, yet the actual message would contain | * |
| | * "COMSUBRON FOURTEEN" in the addressee line. | * |
| **altitude** | = { number } | |
| | * positive integer indicating a platform's position, given as | * |
| | * the number of feet above sea level | * |
| **archive_flag** | = { C \| N \| A \| S } | |
| | * an identifier which identifies the relative age of a given track | * |
| | * tuple, for use data storage | * |
| | * C = current or most-recent track | * |
| | * N = not current track | * |
| | * A = archived (very old) | * |
| | * S = track superseded | * |
| **archive_setup** | = { ALL \| OWNSHIP \| {link_ID} } | |
| | * an initialization command which sets the archive flag within | * |
| | * the message librarian that indicates what messages to archive | * |
| **archive_setup_defaults** | = archive_setup | |
| | * at the time of system start-up, some archive system default | * |
| | * values must be provided | * |

| | | |
|---|---|---|
| **archive_timeout** | = AIR + { number } + SURFACE + { number } + SUBSURFACE + { number } | |
| | * the period of time (in seconds) when a track is no longer of | * |
| | * interest, dropped from the active database, and archived | * |
| **azimuth** | = { number } | |
| | * real, range between 0.0 and 360.0 or range between | * |
| | * -180.0 and 180.0 indicating relative angular bearing within | * |
| | * horizontal plane | * |
| **change_database_ request** | = { ADD_TRACK_TUPLE \| DELETE_TRACK_TUPLE \| UPDATE_TRACK_TUPLE } | |
| | * a message which is sent to the track database manager, | * |
| | * indicating the desired actions to be performed upon the | * |
| | * given data, of the form: | * |
| | * origin + { ADD \| DELETE \| UPDATE } + | * |
| | * { track_ID \| track_tuple } | * |
| **channel_ID** | = { string \| number } | |
| | * given an unique communications system, there will often be | * |
| | * a number of channels over which it is capable of transmitting | * |
| **classification** | = { U \| C \| S \| TS } | |
| | * standard DoD message classification notation | * |
| | * U = UNCLASSIFIED | * |
| | * C = CONFIDENTIAL | * |
| | * S = SECRET | * |
| | * TS = TOP SECRET | * |
| **communications_ message** | = * a file, binary or character oriented, that can be passed | * |
| | * between the Generic C3I Workstation system and a | * |
| | * given communications device. | * |
| **contact_data** | = origin + time + azimuth + (elevation)+ (range) + (velocity) | |
| | * a contact is a object detected by organic sensors | * |

**contact_line** = { string }
* textual portion of a communications message that provides *
* contact related information *

**control_characters** = * standard ASCII non-displayed characters (e.g., carriage *
* return, line feed, new line, etc.) *

**course** = { number } + (T)
* the direction ownship is pointed toward, units between 0° *
* and 360° relative to True North (T) *

**database_request** = (origin) + { string } + time
* an SQL-like command which yields a set of track tuples that *
* are members of the set which meet the specified criteria *

**delete_track_call** = track_ID

**delete_track_tuple** = origin + DELETE + track_ID

**depth** = { number }
* negative integer indicating a platform's position, given as *
* the number of feet below sea level *

**desired_classes** = { track_class + ( range ) }
* implementation could consist of an array of booleans *
* (or booleans and numbers) used to indicate to the Track *
* Filter what tracks to enter into the system *

**desired_format** = message_format
* an identifier which delineates the format that a given message *
* should be (e.g., "JTIDS SITREP") *

**display_tracks_call** = database_request + (refresh_rate)

**edit_command** = { string | desired_format | end_of_message |
database_request }
* specific commands provided to the Message Generator for the *
* purposes of constructing a communications message *

**edit_prompt** = * prompts generated by the Message Generator, asking *
* for specific user input: string input, numeric input, *
* database request input, etc. *

**electronic_mail** = message
* a local header will be used, and any network related routing *
* information will be provided in the body of the text *

**elevation** = { number }
* real, range between -90.0 and 90.0 indicating relative angular *
* bearing within vertical plane *

**emergency_weapon_status_report** = weapon_status + { string }
* a priority weapon status report that notifies the battle manager *
* of a severe change in weapon status: *
* "weapon out of ammunition", "weapon damaged" *
* "weapon jammed", or the like *

**emissions_control_command** = { EMCON | RESTRICTED | UNRESTRICTED }
* a command that initializes a transmission permitted data table, *
* which is to be checked by the comms network monitor and *
* control function *
* EMCON = no transmissions *
* RESTRICTED = limited transmissions *
* UNRESTRICTED = unlimited transmissions *

**end_of_message** = * command signaling the completion of text editing *

**file_request** = origin + filename + (string)
* the user (origin) requesting a particular file may also provide *
* a database identifier in order to reduce search time *

| | | |
|---|---|---|
| **filename** | = { string } | |
| | * a valid operating system file name naming convention should | * |
| | * provide information related to the origin and contents of the | * |
| | * text file | * |
| **format_error** | = { string } + (message) | |
| | * this error message is intended to convey information to the | * |
| | * user about an error made in the format of a message | * |
| **format_request** | = message_type | |
| **geometric_ display** | = {graphical_image} | |
| | * a graphical image will be system specific | * |
| **geometric_ display_request** | = { string } | |
| | * for frames of reference and navigational aides, the user | * |
| | * should be able to overlay geometric shapes and lines onto | * |
| | * the graphics display. | * |
| **graphical_image** | = * a graphical image will be implementation dependent and | * |
| | * include any system commands necessary to produce a | * |
| | * graphical screen display | * |
| **graphics_display** | = {graphical image} | |
| | * this will be the graphical image resulting from the overlaying | * |
| | * of geometric displays, track displays & map displays | * |
| **graphics_track** | = track_ID + latitude + longitude + course + velocity + IFF_class + track_class + observation_time | |
| | * all of the information needed in order to display an NTDS | * |
| | * track symbol (currently two dimensional) * | |
| **header** | = (output_format) + classification + precedence + sender + { addressee \| alias } + (via_line) + (info_line) + (subj_line) | |
| | * the header contains the information most common to all | * |
| | * U.S. Naval textual messages | * |

**hours** = { number }
* integer, range between 0 and 23 *

**IFF_class** = { FRIENDLY | HOSTILE | NEUTRAL | UNKNOWN ... }

**individual_status_report** = origin + weapon_status + ({ string })
* a weapons system status report which indicates weapon *
* loadout *

**info_line** = { string }
* similar to a CC (carbon copy) line in business memorandum, *
* this line contains a list of additional addressees for the *
* given message *

**initiate_transmission_sequence** = link_ID + update_rate + (desired_format) + database_request
* a "startup" message which tells the track report generator to *
* commence message generation, and whether ownship is a *
* participating unit or a master control unit in the given *
* communications network *

**intelligence_data** = { string }
* additional amplifying information that a given sensor may *
* provide concerning the number, type, mission, intent, or *
* actions of a given contact or track *

**intelligence_report** = origin + intelligence_data
* additional amplifying track information provided by platform *
* sensors, that does not directly fit into a track tuple and/or *
* needs to be analyzed to determine its correlation to track data *

**latitude** = { number }
* real, range between -90.0 and 90.0, positive is North, *
* negative is South *

| | | |
|---|---|---|
| **link_ID** | = { LINK4A \| LINK11 \| LINK16 \| OTCIXS \| ... } + ( channel_ID ) | |
| | * an unique identifier, designating the hardware system which | * |
| | * transmits and/or receives digital or analog signals for the | * |
| | * purpose of information transfer | * |
| **loadout** | = { number } + { string } | |
| | * quantity of a given weapon type, for example: | * |
| | * 100 5in_rounds | * |
| | * 38 SM-2ER | * |
| | * 12 MK-48 | * |
| **local_receive_order** | = origin + text_file | |
| **local_track_ID** | = { string } | |
| | * an unique alphanumeric identifier assigned to a track by | * |
| | * ownship sensors, not necessarily the same as an NTDS | * |
| | * or other track ID * | |
| **local_track_information** | = origin + local_track_ID + time + azimuth + (elevation) + (range) + (velocity) | |
| | * track information provided by ownship sensors | * |
| **local_transmit_order** | = ({ string }) + text_file | |
| | * the command sent to a specific link interface | * |
| | * indicating requisite transmission information | * |
| **longitude** | = { number } | |
| | * real, range between -180 and 180.0, positive is East, | * |
| | * negative is West | * |
| **map** | = {graphical image} | |
| | * a map will be a map in the conventional sense, producing an | * |
| | * image signifving geographic and political boundaries, | * |
| | * shoreline information and possibly water depth information | * |

**map_request** = { string } + (refresh_rate)

* a map request will include one of the following: *
*   a) absolute geographic boundaries *
*     (South of w, North of x, East of y, West of z) *
*   b) reference point (latitude, longitude), *
*     relative direction (046T), speed (300KTS) and *
*     orientation (Top = North) *
*   c) perspective (latitude, longitude, altitude) *
*     orientation (Top = North), focus point (lat, long) *
* *
* since windows will be rectangular in nature, mapping *
* information will be provided to fill the entire windowing *
* region *

**maximum_**
**track_number** = { number }

* integer, indicating the upper limit of the quantity of tracks *
* to be maintained by the database system *

**message** = ( header )+ message_body + routing_line

* a message may be treated entirely as one text file,provided *
* that it follows the a standard formatting convention (header *
* first, message body next, routing line last). *

**message_body** = text file

**message_format** = { OTH-T_GOLD | ...} + message_type

**message_template** = message

* a text file containing NULL fields, to be filled in later *

**message_type** = { CONTACT_REPORT | SHORT_CONTACT_REPORT | OPNOTE | GROUP_TRACK_MESSAGES | OVERLAYS | AOI |FOTC_SITREP | QUERY }

* an identifier which specifies what kind of information is *
* being sent in a given message *

**milliseconds** = { number }

* integer, range between 0 and 999 *

**minutes** = { number }
* integer, range between 0 and 59 *

**modify_track_call** = track_tuple

**monitor_mode** = { AUTOMATIC | ADVISE | OFF }
* track database monitor flag that signifies which *
* mode of operation it should operate under: *
* AUTOMATIC -- match, merge, correlate *
* delete and update tracks without *
* user input *
* ADVISE -- send user notification of *
* track ambiguities and constraint *
* violations, provide suggested *
* decisions and defaults, *
* take NO actions *
* await approval prior to taking action *
* OFF -- do not monitor track database *

**monitor_range** = AIR + { number } + SURFACE + { number } + SUBSURFACE + { number }
* the range in nautical miles beyond which a track may be *
* dropped from the active database *

**net_control_flag** = { M | P | A }
* a flag indicating the role of a communications *
* platform within a particular net *
* M = Master Unit *
* P = Participating Unit *
* A = Alternate Mast. Unit *

**network_setup** = { link_ID + { addressee + net_control_flag } } + ({ via_line })
* a message which informs the Comms Network Monitor *
* and Control function of network participants, Master Units, *
* Participating Units, and any additional link related *
* information that will impact system behavior *

**new_message_ notice** = { string }
* a notification announcing the arrival of a new and unread *
* message *

**observer** = { string }
* the name of the platform from which track data is reported *

**observation_time** = time
* the GMT when the contact or track was confirmed *

**origin** = { string }
* the name of the source of providing information *

**output_format** = { O | A }
* when routing a message to a terminal or printer *
* this information will assist in using the appropriate *
* style and font. *
* O = Optical character reader *
* A = All others (default) *

**ownship_ navigation_ information** = course + velocity + latitude + longitude + ({ altitude | depth }) + time

**periodic_ transmission_ command** = network_ID + text_file + refresh_rate
* this is an automatic text file generation which updates *
* its information periodically for routine transmission *

**platform_ status_call** = status_query + refresh_rate

**precedence** = { R | P | O | Z } + (number)
* this field will assist in prioritizing the processing and passing *
* of messages. *
* R = ROUTINE *
* P = PRIORITY *
* O = IMMEDIATE *
* Z = FLASH *
* (If desired, a special access number could be required to be *
* entered by the operator to transmit any message higher than *
* a specified category.) *

**range** = { number }
* distance from ownship, measured in nautical miles *

**read_message_call** = { string }
* a file or message identification command *

**reception_notification** = origin + text_file
* a notification to the system upon the reception of a new *
* data transmission *

**refresh_rate** = { number }
* the frequency with which the specified information is *
* updated (measured in Hertz) *

**relay_command** = link_ID + text_file + (source_format) + (desired_format)
* textual portion of a communications message that provides *
* contact related information *

**report_data** = { string }
* distilled track tuple information, for ready inclusion into a *
* message template *

**reporting_setup** = message_template + (refresh_rate)

**resolution_notice** = { string } + {track_tuple}
* an interrupt message that notifies the track manager of an *
* irregularity or discrepancy in the track database and prompts *
* the track manager for a solution to the resolution (a default *
* solution will always be provided) *

**routing_line** = { link_ID }
* this line specifies what radio equipment & channels are to be *
* used for transmission of a given message *

**save_command** = filename + text_file

**seconds** = { number }
* integer, range between 0 and 59 *
* fractions of seconds may be provided for through the *
* use of milliseconds, or real number representation *
* (0.0 to 59.999..) *

**sender** = { string }
* the name and address of the sender of a message *

**sensor** = * any device capable of detecting the location, direction *
* or characteristics of a track. Most commonly: *
* Radar, Sonar, IRST, ESM, etc. *

**sensor_information** = ASCII_file
* a discrete output file produced by a given sensor that *
* provided track and/or intelligence data *

**set_archive_timeout** = C + { number } + N + { number } + A + { number }
* number is given in seconds *
* system constraints for delineating how old a track must be *
* before its active_flag is changed and subsequent storage *
* considerations are varied *

**set_monitor_ constraints** = set_archive_timeout + set_monitor_range +

set_monitor_mode + set_refresh_rate

* initial set-up constraints for the track database monitor *

**set_monitor_ mode** = monitor_mode

**set_monitor_ range** = AIR + { number } + SURFACE + { number } + SUBSURFACE + { number }
* number is given in nautical miles from ownship tracks *
* reported beyond these ranges are not entered into the track *
* database *

**set_refresh_rate** = { number }
* the frequency of information update, indicated by once every *
* *number* seconds *

**set_track_filter** = maximum_track_number + desired classes
* a message by which the Track Manager may initialize a filter *
* that will limit either the number or type of tracks entering *
* the track database *

**source_format** = message_format
* an identifier which delineates the format of a given message *
* (e.g., "OTH-T GOLD SITREP") *

**status_query** = * a periodic command to update weapon system information *

**status_report** = origin + weapon_status + { string }
* a message which provides current information about a given *
* weapon system, including loadout *

**status_update_request** = { string }
* a message requesting system status, or specific system *
* information *

**string** = * any combination of upper and lower case letters, *
* numbers and standard VT100 keyboard characters *
* (e.g., #, *, @, %, $, &, ! , ~, ", |, \ / , etc.) *

**subj_line** = { string }
* a subject line associated with a message, usually *
* summary or topical in nature *

**terminal_input** = * any form of data entry including: keypad, mouse, *
* trackball, verbal, variable action display, or the like *
* capable of registering the user's choice of action, *
* acknowledgement, selection, or entry of numeric *
* and textual information. *
* Input information will include: *
* transmission sequence data *
* archive setup data *
* track monitor data *
* track filter data *
* add track data *
* track update data *
* delete track data *
* status input *
* track display data *
* edit data *
* read message data *
* reporting data *
* network data *
* emissions data *
* data-less input used for specifying a user's choice *
* or preference includes: *
* archive setup option *
* monitor constraints option *
* track filter option *
* display tracks option *
* add track option *
* update track option *
* delete track option *
* display status option *
* display track option *
* generate message option *
* view message option *

| | | |
|---|---|---|
| | * network setup option | * |
| | * emissions status option | * |
| | * set reporting option | * |
| | * initialize transmission option | * |
| | * cancel option | * |

**terminal_output** = * any combination of textual and/or graphical display *
* that is supported by a windowing environment *
* Output information will include: *
* resolution screen *
* intelligence report screen *
* track tuples screen *
* emergency status screen *
* message arrival notification *
* weapon status screen *
* graphics display screen *
* edit screen *
* text screen *

**text_file** = { string | control_characters }
* a file as defined by a given operating system, and containing *
* containing only valid ASCII characters *

**time** = hours + minutes + seconds + (milliseconds) + ({ Z | L })
* time used for reporting purposes will be given relative to *
* Greenwich Mean Time -- aka Zulu Time (Z) as opposed to *
* local time (L) *

**track** = track_ID + { observer | origin } + observation_time +
track_class + IFF_class + latitude + longitude +
({ altitude | depth }) + course + velocity + ({ string })
* this is a working subset of all available track data *

**track_class** = { SURFACE | SUBSURFACE | AIR }

**track_display_call** = database_request + refresh_rate

**track_filter_ defaults** = set_track_filter
* at the time of system start-up, some default values must *
* be provided to determine what messages to process and *
* what messages to ignore *

**track_ID** = { string }
* an unique 8 character identifier, intended to serve as a key field *

**track_line** = { string }
* textual portion of a communications message that provides *
* track related information *

**track_monitor_ defaults** = set_monitor_constraints
* at the time of system start-up, default values will be provided *
* for the track database monitor *

**track_request** = database_request + refresh_rate

**track_tuple** = ( origin )+ track + archive_flag
* the information stored in the track database *

**translation_ command** = message + (source_format) + (desired_format)
* the command which specifies the format of the given text file, *
* as well as the desired new format after the resulting translation *

**transmission_ command** = message
* the specific information necessary to commence digital *
* transmission of the given text file over a he communications *
* tlink designated in that message's routing line *

**transmission_ sequence_ defaults** = message_template + refresh_rate

| | |
|---|---|
| **transmit_ command** | = link_ID + message<br>* if the text file contains an outgoing formatted message then *<br>* the addressee and the communications_link may be inferred, *<br>* otherwise (addressee) will be used for determining to whom *<br>* an unformatted text_file should be sent, (communications *<br>* link) will only need to specify a particular link is to be used *<br>* if other than default * |
| **update_ track_tuple** | = origin + UPDATE + track_tuple |
| **via_line** | = { string }<br>* commonly, this line would contain a list of addressees that *<br>* are directly in the chain of command between the sender and *<br>* the intended recipient * |
| **velocity** | = { number }<br>* the speed of a given track or contact, measured in knots *<br>* (i.e., nautical miles per hour) * |
| **weapon_status** | = origin + [ DAMAGED \| RELOADING \|<br>LAUNCHING \| READY \| SERVICE_REQUIRED \|<br>SLEWING \| OUT_OF_AMMUNITION \|<br>SECURED \| MAINTENANCE \| ENGAGING ] +<br>{loadout}<br>* the category of preparedness associated with a given weapon *<br>* system * |
| **window** | = * a system window will be implementation dependent and *<br>* include any system commands necessary to produce a *<br>* window display * |
| **window_request** | = { string }<br>* a window request must provide information concerning size *<br>* and screen location, as well as the type of window it will be, *<br>* textual, interactive, graphical, color, etc. * |

# APPENDIX F

## ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| AAW | Anti-Air Warfare |
| AAWC | Anti-Air Warfare Commander |
| ACDS | Advanced Combat Direction System |
| ASW | Anti-Submarine Warfare |
| ASWC | Anti-Submarine Warfare Commander |
| ASuW | Anti-Surface Warfare |
| ASuWC | Anti-Surface Warfare Commander |
| BBN | Bolt, Beranek and Newman, Inc. |
| BDA | Battle Damage Assessment |
| C2 | Command and Control |
| C3 | Command, Control & Communications |
| C3I | Command, Control, Communications & Intelligence |
| C&D | Command and Decision |
| CAPS | Computer Aided Prototyping System |
| CDS | Combat Direction System |
| CINC | Commander-IN-Chief |
| CIWS | Close In Weapon System |
| CPU | Central Processing Unit |
| CWC | Composite Warfare Commander |
| ESM | Electronic warfare Support Measures |
| EW | Electronic Warfare |
| EWC | Electronic Warfare Coordinator/Commander |
| FCS | Fire Control System |
| FOTC | Force Over-the-horizon Track Coordinator |
| GFCS | Gun Fire Control System |
| GMT | Greenwich Mean Time |
| GPS | Global Positioning System |
| GWS | Gun Weapon System |

| | |
|---|---|
| IDS | Interface Design Specification |
| IEEE | Institute for Electrical and Electronics Engineering |
| IFF | Indentify: Friend or Foe |
| IFFN | Indentify: Friend, Foe or Neutral |
| INS | Inertial Navigation System |
| IR | Infrared |
| IRST | InfraredSearch and Target Designation |
| ISAR | Inverse Synthetic Aperture Radar |
| | |
| JCS | Joint Chiefs of Staff |
| JOTS | Joint Operational Tactical System |
| JTIDS | Joint Tactical Information Distribution System |
| | |
| LOB | Line Of Bearing |
| | |
| NASA | National Aeronautics and Space Administration |
| NGCR | Next Generation Computer Resources |
| | |
| met | maximum execution time |
| Mk 45 | 5"/54 gun mount |
| Mk 46 | Lightweight torpedo |
| Mk 86 | Digital gun fire control system |
| Mk 116 | Surface ship underwater fire control system |
| ms | millisecond (.001 second) |
| | |
| NCA | National Command Authority |
| NCCSA | Navy Command and Control System, Afloat |
| NTDS | Naval Tactical Data System |
| | |
| OTC | Officer in Tactical Command |
| OTCIXS | Officer in Tactical Command Information eXchange Sytstem |
| OTH-T | Over The Horizon - Targeting |
| | |
| PPI | Plan-Position Inidicator (e.g., a two dimensional radar display screen) |
| PSDL | Prototype System Description Language |
| | |
| RADAR | RAdio Detection And Ranging |

| | |
|---|---|
| SAF | Infrared search and target designation system, developed by GE/SPAR |
| secs | seconds |
| SITREP | SITuation REPort |
| SLQ-32 | ESM device capable of detecting/categorizing threats and providing ECM , developed by Raytheon |
| STW | STrike Warfare (aka, offensive land attack) |
| STWC | Strike Warfare Commander |
| SPY-1 | S-band phased array radar, developed by RCA (now GE) |
| SQS-53C | bow mounted long range low-frequency sonar system, developed by GE |
| TADIL | TActical Digital Information Link |
| TFCC | Tactical Flag Command Center |
| TWCS | Tomahawk Weapons Control System |
| UFCS | Underwater Fire Control System |
| WCS | Weapon Control System |
| WMA | Warfare Mission Area |
| WS | Weapon System |

## REFERENCE LIST


1. Berzins, V. and Luqi, *Software Engineering with Abstractions*, Addison-Wesley Publishers, Reading, MA, 1991.
2. Booch, Grady, *Software Components with Ada*, Second edition, Benjamin/Cummings, Menlo Park, CA, 1987.
3. Friedman, Norman, *The Naval Institute Guide to World Naval Weapons Systems*, Naval Institute Press, Annapolis, MD, 1989.
4. Galdorisi, George V., "Making it Work", *U.S. Naval Institute Proceedings*, September, 1990.
5. Greif, Alan and McKnight, Clarance, "A New Picture of the Battlefield", *Aerospace America*, May 1990.
6. Hughes, Wayne P., *Fleet Tactics*, Naval Institute Press, Annapolis, MD, 1987.
7. Hwang, J., et al., editors, *Selected Analytical Concepts in Command and Control*, Gordon and Breach Science Publishers, New York, NY, 1982.
8. Loescher, Michael S., "Navy Reshapes, Develops Copernicus Architecture", *Signal*, July 1990.
9. Luqi, Unpublished memorandum, 1989.
10. Luqi, and Berzins, V.,"Rapidly Prototyping Real-Time Systems", *IEEE Transactions on Software Engineering*, September 1988.
11. Luqi, Berzins, V., and Yeh, R., "A Prototyping Language for Real Time Software", *IEEE Transactions on Software Engineering*, October 1988.
12. Luqi, and Ketabchi, M., "A Computer-Aided Prototyping System", *IEEE Transactions on Software Engineering*, March 1988.
13. Orr, George E., *Combat Operations $C^3I$: Fundamentals and Interactions*, Air University Press, Maxwell Air Force Base, AL, 1983.
14. Parker, Jay P., "$C^2$ for the Cutting Edge", *U.S. Naval Institute Proceedings*, October, 1988.
15. Parker, Jay P., "CDS: The Combat System Trigger", *U.S. Naval Institute Proceedings*, October, 1988.
16. Rackham, Peter, editor, *Jane's $C^3I$ Systems*, Jane's Information Group, Inc., Alexandria, VA, 1989.

17. Royce, Winston W., "Managing the Development of Large Software Systems," *IEEE Proceedings*, IEEE Wescon, August 1970.

18. Schweiger, Jeff, "Development of a Composite Warfare Commander (CWC) Concept $C^2$ Architecture", Unpublished report, 21 June 1990.

19. Seveney, James A., and Steinberg, Guenter P., *Requirements Analysis for a Low Cost Combat Direction System*, Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1990.

20. Shimeall, Timothy J., "TWIRP: Testing Within Iterative Rapid Prototyping", NPS-52-90-030, Computer Science Department, Naval Postgraduate School, Monterey, CA, July 1990.

21. Shneiderman, Ben, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley Publishing Company, Reading, MA, 1987.

22. Tannenbaum, Andrew S., *Computer Networks*, Second edition, Prentice Hall, Englewood Cliffs, NJ, 1988.

23. Van Cott, Harold P. and Kinkade, Robert G., editors, *Human Engineering Guide to Equipment Design*, Sponsored by Joint Army-Navy-Air Force Steering Committee, American Institutes for Research, Washington, DC, 1972.

24. Voss, Philip D., "Battle Force ASW: $M^3$", *U.S. Naval Institute Proceedings*, January, 1989.

25. White, Laura, *The Development of a Rapid Prototyping Environment*, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1989.

26. Yourdon, Edward, *Modern Structured Analysis*, Yourdon Press, Englewood Cliffs, NJ, 1989.

27. By the Staff of Carrier Group Seven, U.S. Pacific Fleet, "Tomahawk Equals True Value", *U.S. Naval Institute Proceedings*, September, 1990.

28. By the Staff of Destroyer Squadron Twenty-One, "Changing World, Changing Warfare", *U.S. Naval Institute Proceedings*, September, 1990.

29. MIL-STD-1472D, *Human Engineering Design Criteria for Military Systems, Equipment, and Facilities*, 14 March 1989.

30. PDW 120-S-00533 (Rev.B, Change 4), "Over-the-Horizon Targeting (OTH-T) Gold Reporting Format", Naval Tactical Interoperability Support Activity, 30 June 1989.

31. "Vice Admiral Hogg on Navy $C^3I$", *The $C^3I$ Handbook*, EW Communications, Inc., Third Edition, 1986.

32. *Department of Defense Dictionary of Military and Associated Terms*, Joint Chiefs of Staff Publication Number 1, 1 June 1979.
33. Naval Research Advisory Committee, "Next Generation Computer Resources", Committee Report, February, 1989.
34. *System V Enhancements Overview*, Part Number 800-1541-03, Sun Microsystems®, 19 September 1986.
35. Commander, Space and Naval Warfare Systems Command, "Review of Survivable Adaptable Fiber Optic Embedded Network (SAFENET) Local Area Network (LAN) Standard and Handbook", memorandum, Enclosure (3), SAFENET I Executive Summary, 19 June 1990.
36. Gauss, John, Afloat Command and Control Briefing, 23 May 1990.
37. Schweiger, Jeffrey M., C3I Architecture Briefing, 31 August 1990.